

SUPER

Anno 2
n° 4 - Aprile 1985

L. 7.500

Spedizione in
abbonamento
postale Gruppo III/70

4

SUPERCOMMODORE
la rivista per gli utenti
dei prodotti **COMMODORE**

C- COMMODORE

Una pubblicazione della

J. soft EDITRICE
in collaborazione con

GRUPPO
EDITORIALE
JACKSON



LIGHTSAVER

**CACCIA AL TESORO
PER VIC 20**

SUPERTANK PER C64

**OTHELLO PER
C16**

Dalla grande edicola Jackson

Tutto sul personal computer

APRILE 1985
Personal
La prima rivista indipendente per gli utenti Olivetti

Personal
in salopette,
elettrodomestico
è salvo
Toga per
un personal
M10
senza
veli



PERSONAL O

In questo numero:
Sci e videotex

Personal
in salopette,
l'elettrodomestico è salvo

Scheda telematica
su M24

M10 senza veli

Toga per un personal

BIT

In questo numero:
**Speciale
Grafica**

Bitest:
VDS ECO 3

In prova:
Sharp PC-1260
Copal SC-1200

Softest: Jane



PC MAGAZINE

In questo numero:
Pc Network
Banche dati: The Source
Ca-Executive
L'illusione delle interfacce ...



PERSONAL SOFTWARE

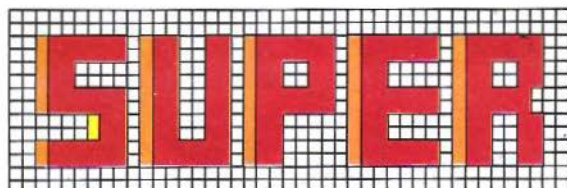
In questo numero:
Equadif per C64
Basket per Spectrum
**Battaglia del lago
ghiacciato per C16**
Totocalcio per Apple

Personal-O/PC/Bit/Personal Software
sono pubblicazioni firmate:

GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano





5 EDITORIALE POINT

di **Pietro Dell'Orco**

7 READ & WRITE LA POSTA DEI LETTORI

9 LA CULTURA IS INFORMATICA ...then ABC DEL LINGUAGGIO MACCHINA prima parte di **U.G. Barzaghi**

24 REM:HW IL JOYSTICK terza parte di **U.G. Barzaghi**

31 LOAD 31 OTHELLO PER C16 di **A. Motta**

34 INDICE DI CASSETTA di **P.N. Peasley** trad. e adatt. di **F. Sarcina**

38 STAMPARE LO SCHERMO AD ALTA RISOLUZIONE di **G. Peele** trad. e adatt. di **F. Sarcina**

42 LA STATISTICA DIVENTA FACILE di **A. Burke Luitich** trad. e adatt. di **F. Sarcina**

48 LIGHTSAVER di **J. Rhees** trad. e adatt. di **S. Albarelli** e **M. Anticoli**

56 VERSO LE INDIE di **M.J. Winter** trad. e adatt. di **F. Sarcina**

63 CACCIA AL TESORO PER VIC 20 di **S. Albarelli**

66 SUPERTANK PER C64 di **B. Litinsky** trad. e adatt. di **S. Albarelli** e **M. Anticoli**

71 VICsound A TUTTO BACI! PER C64 di **A. e M. Fossati**

78 F I N T RECENSIONI LIBRI di **M. Cristuib Grizzi**

80 input-output PICCOLI ANNUNCI

J. soft s.r.l.
**DIREZIONE, REDAZIONE,
AMMINISTRAZIONE**
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.88.228

DIRETTORE RESPONSABILE:
Pietro Dell'Orco

COORDINAMENTO TECNICO:
Riccardo Paolillo

REDAZIONE:
Mauro Cristuib Grizzi

**HANNO COLLABORATO A
QUESTO NUMERO**
S. Albarelli
M. Anticoli
U.G. Barzaghi
A. e M. Fossati
A. Motta
F. Sarcina

GRAFICA E IMPAGINAZIONE:
Margherita La Noce
Ivana Rossi
Raffaella Toffolatti

FOTOCOPOSIZIONE:
Graphotek - Via Astesani, 16
Tel. 02/64.80.397
20161 MILANO

CONTABILITÀ:
Giulia Pedrazzini
Flavia Bonatti

**AUTORIZZAZIONE ALLA
PUBBLICAZIONE:**
Tribunale di Milano n° 201
del 14.04.1984

STAMPA:
Litografia del Sole
Albairate (MI)

PUBBLICITÀ
Concessionaria per l'Italia e
l'Estero J. Advertising s.r.l.
Viale Restelli, 5
20124 MILANO
Tel. (02) 68.82.895-68.80.606-68.87.233
Tlx. 316213 REINA I

Concessionaria esclusiva per la
DIFFUSIONE in Italia e Estero:
SODIP - Via Zuvetti, 25
20125 MILANO

Spedizione in abbonamento
postale Gruppo III/70
Prezzo della rivista L. 7.500
Numero arretrato L. 15.000
Abbonamento annuo: L. 82.500
(11 numeri con cassetta).
L. 38.500 (11 numeri senza cassetta).
per l'estero: L. 110.000 (11 numeri con
cassetta) - L. 50.000 (11 numeri senza
cassetta)

I versamenti devono essere indirizzati a:
J. soft s.r.l.
Viale Restelli, 5 - 20124 MILANO
mediante emissione di
assegno bancario, cartolina
vaglia o utilizzando
il c/c postale n. 19445204.
Per i cambi di indirizzo indicare,
oltre al nuovo, anche l'indirizzo
precedente ed allegare alla
comunicazione l'importo di L. 500,
anche in francobolli

© TUTTI I DIRITTI DI
RIPRODUZIONE O TRADUZIONE
DEGLI ARTICOLI PUBBLICATI
SONO RISERVATI

GLI ARTICOLI TRADOTTI SONO
TRATTI DALLA RIVISTA COMPUTE!
O DA COMPUTE! GAZZETTE -
COMPUTE! PUBLICATION, INC.
P.O. BOX 5406,
GREENSBORO, NC 27403 - USA



CERCA:

ambosessi di qualsiasi età, residenti in Italia
o all'estero

REQUISITI RICHIESTI:

forte interesse per gli home computer Commodore
disponibilità tempo libero per appassionante
lettura di "superivista" dedicata

OFFRE:

abbonamento a 11 numeri di SuperCommodore
al **prezzo speciale di**
L. 66.000
compresa cassetta
oppure
L. 30.000
per la sola rivista

inviando il coupon pubblicato a fondo pagina otterrete pronta soddisfazione alla vostra richiesta.

Abbonarsi è semplice! Effettuate il versamento con l'apposito modulo c.c.p. inserito in questo fascicolo, oppure
ritagliate il tagliando abbonamenti pubblicato in questa pagina e spedite lo allegando un assegno intestato a:
J. soft - Viale Restelli, 5 - 20124 Milano.

Tagliando abbonamento a SUPECCOMMODORE da inviare in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano

- ☐ Abbonamento a 11 numeri di SUPERCOMMODORE al prezzo speciale di L. 30.000
- ☐ Abbonamento a 11 numeri di SUPERCOMMODORE + cassetta con tutti i programmi pubblicati al
prezzo speciale di L. 66.000

cognome _____ nome _____
via _____ città _____
cap. _____ provincia _____ data _____
firma _____

POINT



Fare!

Da sempre l'uomo è in cerca di nuove esperienze. Quella dell'informatica è forse una delle più affascinanti. Ovviamente tale esperienza dev'essere affrontata nel modo giusto.

Oggi tutti si sforzano di dare consigli su come muoversi nell'apparentemente intricato mondo dei calcolatori. Non è necessario essere esperti matematici oppure piccoli geni per avvicinare la materia.

Fortunatamente c'è chi, come sempre, pensa e provvede anche per chi non ha mezzi o capacità di fare in proprio. Ecco dunque spiegata la nascita di miriade di pubblicazioni, la disponibilità di centinaia di programmi pronti all'uso e via dicendo.

Forse il discorso è banale, ma provando a riflettere si constaterà come tutto ciò abbia contribuito in maniera determinante alla diffusione del personal e home computer. Si sa l'hardware senza il software non ha storia. Il problema, per l'utente medio, è quello di non fossilizzarsi solo perché c'è comunque qualcuno che ci pensa. La miglior politica è quella di considerare tutto come un aiuto per imparare a far da sé. Solo in questo modo si potrà procedere, magari a piccoli passi, fino a padroneggiare in qualche modo la situazione.

Ogni nuova scoperta deve servire da pretesto per cimentarsi; aspettare che la soluzione di un problema specifico arrivi dalle pagine di una rivista o dalla commercializzazione di un nuovo prodotto non è certo la strada giusta.

L'attesa passiva non farà vivere l'esperienza con il computer nel modo corretto.

Dunque sotto! Ogni riga di programma dev'essere uno spunto, la routine "wyz" deve servire a risolvere quel problema che vi assilla da giorni.

La parola d'ordine è quindi "fare" poi, per quelli che riescono meglio, c'è la possibilità di inviare i propri lavori in redazione per l'eventuale pubblicazione sulla rivista che, oltre alla soddisfazione, può far guadagnare qualche lira.

Pietro Dell'Orco

Per collaborare a SUPERCOMMODORE

La rivista è interessata ad articoli e programmi riguardanti la linea Commodore, di interesse generale, chiari ed esaurienti. Vi proponiamo una piccola "guida", che ha particolarmente lo scopo di rendere più accurata la stesura dei vostri manoscritti: seguendo i nostri consigli si accresceranno le probabilità che le vostre buone idee e i vostri programmi vengano pubblicati.

1 l'angolo superiore sinistro della prima pagina dovrà contenere: nome, cognome, indirizzo, numero telefonico, codice fiscale, data di spedizione, luogo e data di nascita.

2 l'angolo superiore destro della prima pagina dovrà contenere la marca e il tipo di computer al quale il lavoro si riferisce, unitamente ad eventuali espansioni di memoria o periferiche richieste.

3 il titolo dell'articolo, sottolineato, dovrà iniziare a circa due terzi in altezza della prima pagina.

4 le pagine seguenti potranno essere battute normalmente, con la condizione che l'angolo superiore destro contenga un'abbreviazione del titolo e del cognome, unitamente al numero di pagina. Per esempio, Sprite Ed.../Brambilla/2.

5 il testo dell'articolo dovrà essere battuto a macchina con interlinea di uno spazio e mezzo, massimo due spazi; un margine di almeno un centimetro dovrà essere lasciato su entrambi i lati dello scritto.

6 dovranno essere usati fogli in formato UNI A4 (cm. 21 x 29,7) e il testo, scritto in caratteri maiuscoli e minuscoli, dovrà occupare una sola facciata del foglio.

7 nel caso il testo comprenda più fogli, questi dovranno essere uniti con un fermaglio o con un punto metallico aperto.

8 avendo intenzione di spedire più di un articolo, questi dovranno essere inviati separatamente insieme alla rispettiva copia su supporto magnetico.

9 programmi brevi (meno di 20 linee) potranno essere inseriti nel testo, mentre programmi più lunghi dovranno essere listati separatamente. È **ESSENZIALE** per noi disporre di una copia del programma registrata più volte su supporto magnetico, su entrambi i lati dello stesso. È preferibile usare nastri di buona qualità e di lunghezza non eccessiva; la cassetta o il disco dovranno essere etichettati con il nome dell'autore, il titolo dell'articolo, il computer interessato e soprattutto le eventuali espansioni richieste. Come sug-

gerimenti di programmazione si consiglia di usare le istruzioni "CHR\$(x)", "TAB(x)", "SPC(x)", piuttosto che stringhe di manipolazione del cursore.

Ad esempio, per uno scroll di 5 linee l'istruzione "FOR I = 1 TO 5:PRINT:NEXT I". è molto più interpretabile di 5 Q inverse; e, invece di una dozzina di simboli di cursore a destra, perché non usare semplicemente "PRINT SPC(12)"? Un rapido controllo dei programmi per operare queste sostituzioni sarà molto apprezzato da noi e dai lettori.

10 per maggior chiarezza, all'interno dell'articolo è conveniente usare caratteri maiuscoli riferendosi a istruzioni BASIC (esempio RETURN, LIST, RND, PRINT ecc.). Se si desidera evidenziare una parola, è preferibile sottolinearla piuttosto che scriverla in carattere maiuscolo.

11 gli articoli ed i programmi potranno avere qualsiasi lunghezza: da una routine di una sola linea fino a programmi molto complessi.

12 volendo includere diapositive, queste dovranno avere formato 24x36, o 6x6.

13 non prenderemo in considerazione articoli che siano stati sottoposti ad altre case editrici.

14 il materiale non pubblicato non verrà restituito.

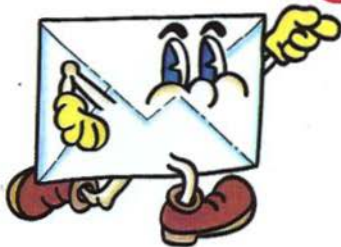
15 il compenso per la collaborazione prestata sarà commisurato alla complessità e all'interesse del testo e/o del programma (da un minimo di L. **50.000** a un massimo di L. **300.000**). Il pagamento verrà effettuato in caso di pubblicazione del lavoro.

Inviare idee e programmi a:
SUPERCOMMODORE
Viale Restelli, 5
20124 Milano

e saremo liettissimi di pubblicare i contributi migliori.

La Redazione

READ & WRITE



La posta
dei lettori

Contatore del nastro e posizione dei programmi

Spett. Redazione,
ho notato con dispiacere che sulle cassette allegate alla vostra rivista non compaiono i numeri relativi alla posizione del contanastro sul registratore per ogni programma registrato, ed inoltre che molti nomi di programma sono diversi da quelli che appaiono sull'etichetta della cassetta. Fornendo il numero del contanastro corrispondente all'inizio di ogni programma facilitereste di molto la ricerca dei programmi più interessanti, non credete?

Franco Mitis

R. Purtroppo esiste un certo numero di versioni del registratore Commodore, dalle più vecchie alle più nuove, ed ognuna di queste è provvista di un contanastro che gira con velocità diversa dalle altre. Il pubblicare per ogni programma i valori del contanastro rilevati su un nostro registratore favorirebbe esclusivamente i possessori della stessa versione di registratore, mentre provocherebbe confusione in chi disponesse di altre versioni dello stesso o addirittura di registratori Commodore-compatibili od anche normali registratori portatili interfacciati al computer.

I nomi dei programmi così come sono incisi sulla cassetta a volte non coincidono con quelli presenti sull'etichetta della stessa, e ciò per un motivo tecnico: i master da cui si ricavano le cassette vengono preparati, per minimizzare il rumore di fondo, su floppy disk. Occorre quindi distinguere le versioni dei programmi per C64 da quelle per VIC 20, dal momento che un singolo floppy disk non può contenere due programmi con lo stesso nome. Il sistema per scoprire subito i nomi attribuiti sulla cassetta è del resto molto semplice: digitando LOAD*** e premendo RETURN il computer cercherà sul nastro un programma di nome * (che naturalmente non

esiste) e quindi passerà tutta la cassetta restituendo i nomi dei programmi con il messaggio FOUND.... Si potrà approfittarne per segnare su un foglio la posizione del contanastro ad ogni apparizione del messaggio FOUND, insieme al nome corretto del programma. Sull'etichetta della cassetta, per non creare confusione, abbiamo ritenuto preferibile riportare i nomi dei programmi così come appaiono negli articoli della rivista.

C64 europeo ed americano

Spett. Redazione,
ho sentito dire che i C64 prodotti in America non funzionano con la corrente italiana. È solo un problema di diverso voltaggio, oppure ci sono differenze nel computer?

Ennio Di Stefano

R. In realtà i computer Commodore — come del resto quasi tutti i computer — vengono prodotti in numerosi stabilimenti sparsi per il mondo, e non solo negli Stati Uniti. Curiosando tra le etichette appiccate su un drive 1541 ci è capitato poco fa di trovare un "Made in Taiwan"... Del resto non ha una grande importanza dove un computer venga prodotto, ma che siano rispettati nella produzione gli standard qualitativi imposti dal progetto originario. È comunque vero che i C64 prodotti per essere venduti sul mercato Nordamericano non funzionano in Europa, e ciò per due fondamentali motivi: la tensione elettrica standard negli U.S.A. è di 110 Volt contro i 220 Volt europei — fatto questo che sarebbe tuttavia facilmente aggirabile con un opportuno trasformatore di tensione —, ma la frequenza di rete è in U.S.A. di 60 Hz contro i nostri 50 Hz. Tale differenza di frequenza richiede una taratura a livello hardware completamente differente.

Il secondo motivo è la differenza di stan-

dard televisivo adottato (NTSC negli U.S.A. e PAL in Europa con l'esclusione della Francia che adotta il SECAM) e quindi diversa regolazione del segnale in uscita dal computer verso il televisore.

Il C64 dispone di una locazione di memoria che fornisce direttamente, con una PEEK, il tipo di versione della macchina (NTSC o PAL). Questa locazione è la 678: se il risultato della PEEK è 0, il computer è tarato per l'NTSC, mentre se è 1, il computer è tarato per il PAL.

L'Avvento dei modem

Spett. Redazione,
ho letto con molto interesse i vostri articoli sul modem, e ne sono rimasto affascinato al punto da cercare di acquistarne uno. Purtroppo non sono riuscito a reperirlo né nella mia città né telefonando altrove. Potete consigliarmi su dove poterne reperire uno?

Francesco Di Salvo

R. Il modem è una periferica veramente in grado di aprire nuovi orizzonti a chi usi il computer per divertimento, hobby o lavoro. In Italia siamo appena all'inizio della "rivoluzione informatica", ed una concreta diffusione dei personal ed home computer è iniziata praticamente solo dall'inizio dello scorso anno. L'utilizzo di un modem per comunicare con altri computer o con centri di informazione presuppone, com'è ovvio, la diffusione di altri computer dotati di questa periferica o l'esistenza di network sul tipo degli statunitensi "CompuServe" o "The source". Occorre perciò attendere che si consumi questo "gap" rispetto agli U.S.A. e che inizi anche da noi la diffusione della pratica della telecomunicazione tra computer. Acquistare ora un modem è un po' come acquistare un'automobile prima che vengano costruite le strade sulle quali utilizzarla...

La piccola guida del principiante

Che cos'è un programma?

Il solo computer non può compiere alcuna operazione. Un computer possiede potenzialità, ma, come una macchina senza benzina, senza un programma non può funzionare. La maggior parte dei programmi pubblicati su **SUPERCOMMODORE** per i computer Commodore sono scritti in un linguaggio per calcolatori chiamato BASIC. Il BASIC è facile da imparare ed è disponibile, di serie, nel VIC 20 e nel Commodore 64.

Programmi BASIC

Ogni mese **SUPERCOMMODORE** pubblica programmi sia per il VIC 20 che per il C64. Tanto per cominciare, se possiedi un VIC 20 copia solo i programmi scritti per la tua macchina, indicati con "versione per VIC 20". Più tardi, quando avrai acquisito esperienza con il BASIC del tuo computer, potrai cercare di copiare e convertire programmi scritti per altri calcolatori. Diversamente dal linguaggio corrente, che può essere variamente interpretato, il BASIC di solito ha un solo modo corretto per indicare qualcosa.

Ogni lettera, carattere o numero ha il suo significato. Un errore banale è costituito dalla sostituzione del numero "0" con la lettera "O" oppure il carattere minuscolo "l" invece del numero "1" o ancora il carattere maiuscolo "B" con il numero "8".

Devi anche inserire tutta la punteggiatura, i due punti (:) e le virgole, copiando esattamente ciò che appare sulla rivista. Gli spazi possono essere importanti. Per essere sicuro copia il listato esattamente come si presenta.

Le parentesi e i caratteri speciali

L'eccezione per questa regola di copiatura si presenta quando incontrerai indicazioni tra parentesi, quali: "{GIU}". Ogni cosa compresa tra parentesi è un carattere speciale oppure un carattere che non può essere facilmente prodotto con la stampante. Incontrando un carattere di questo tipo fai riferimento alla "Piccola guida per l'input dei programmi".

Le istruzioni DATA

Alcuni programmi contengono una sezione, o delle sezioni, di istruzioni DATA. Queste linee di istruzione forniscono le informazioni di cui il programma ha bisogno. In alcuni casi le istruzioni DATA costituiscono il programma vero e proprio, in altri contengono codici grafici. Queste

linee sono particolarmente soggette agli errori. Se un solo numero in una linea di istruzione DATA è sbagliato, il calcolatore potrebbe "plantarsi" o distruggere il programma. La tastiera e il tasto STOP appaiono inattivi e lo schermo completamente vuoto. Non lasciarti prendere dal panico. Non si è verificato alcun danno.

Per riprendere il controllo devi spegnere il computer e successivamente riaccenderlo. Ciò cancellerà qualsiasi programma presente in memoria, per cui è sempre necessario fare il SAVE del tuo programma prima di comandare il RUN.

Se il computer si ferma, puoi caricare (LOAD) il programma e cercare l'errore.

A volte, quando il programma viene "lanciato", un'istruzione DATA errata può causare un messaggio di errore.

Il messaggio di errore potrebbe riferirsi alla linea di programma che legge (READ) il contenuto delle istruzioni DATA.

Come conoscere il computer

Dovresti prendere confidenza con il computer prima di procedere alla copiatura del programma.

Impara le istruzioni che si usano per memorizzare e richiamare i programmi da nastro o da disco. Dovrai conservare una copia del tuo programma, se non vorrai copiarlo ogni volta che lo devi usare. Impara ad usare le funzioni di "editing" della tua macchina. Come puoi correggere un errore? Puoi sempre ricopiare la linea e in questo caso devi sapere come procedere. Sapresti come inserire i caratteri in "inversa", i caratteri minuscoli e quelli di controllo?

Tutto ciò è spiegato nel manuale del calcolatore.

Un veloce ripasso

- 1) Copia il programma una linea alla volta, con ordine. Premi RETURN alla fine di ogni linea. Usa il tasto "DEL" per correggere gli errori.
- 2) Confronta la linea copiata con quella presente nella rivista. Puoi controllare l'intero programma nel caso in cui si presentino un errore quando esegui il RUN.
- 3) Accertati di aver inserito le istruzioni tra parentesi graffe con gli appropriati caratteri di controllo (fai riferimento alla "Piccola guida per l'input dei programmi" che trovi in questa stessa rivista).

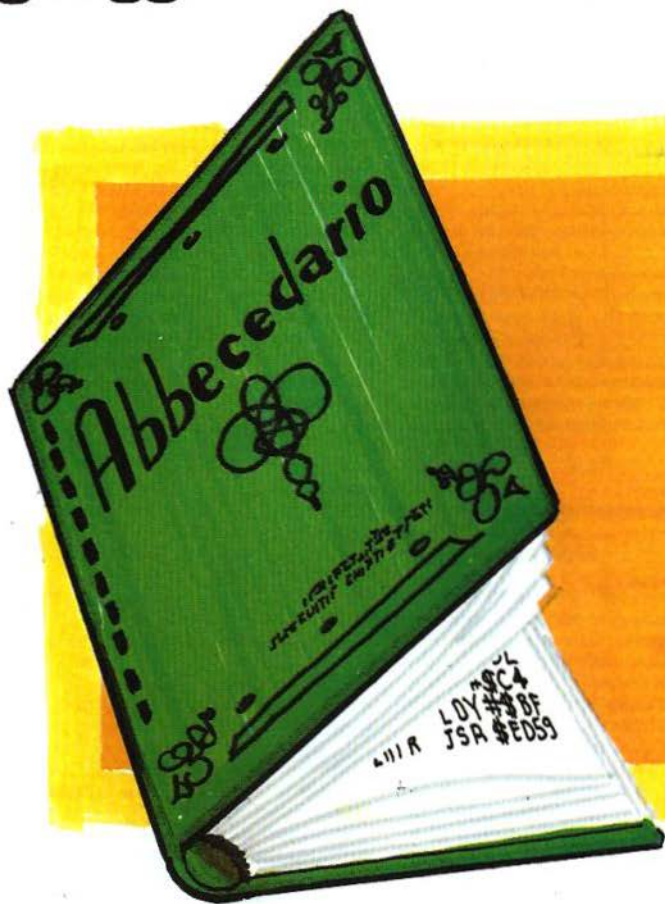
Siamo spiacenti di non poter rispondere singolarmente alle richieste di informazioni circa i programmi, prodotti o servizi apparsi su **SUPERCOMMODORE**.

ABC del Linguaggio macchina: prima parte

di U. G. Barzaghi

La serie di articoli che inizia con questo numero di "SUPERCOM-MODORE" rappresenta un, speriamo utile, complemento alla pubblicazione, sul numero della rivista di Gennaio, del programma SUPERMONITOR (cui, d'ora in avanti, si farà riferimento come SM) per l'assemblaggio, la correzione e l'esecuzione di programmi in linguaggio ASSEMBLY.

Lo scopo di questi articoli è di illustrare, attraverso una serie di esempi pratici e tramite l'uso di SM, le nozioni fondamentali della programmazione in linguaggio macchina, presumendo da parte del lettore una conoscenza minima, o quasi nulla, dell'argomento. Ciò significa che, almeno per i primi articoli, gli argomenti trattati saranno, ovviamente, rappresentati dai soliti elementi di base della programmazione, come la numerazione binaria, il complemento a uno e a due o i vari formati di indirizzamento delle istruzioni in codice mnemonico. Preghiamo pertanto quei lettori che possono vantare già una conoscenza un po' più approfondita della materia di avere la pazienza di attendere il momento in cui il nostro breve "corso" avrà permesso anche agli altri lettori di compiere i primi passi verso una, speriamo, più approfondita comprensione della programmazione in linguaggio macchina. Cercheremo, per quanto ci sarà possibile, di usare un linguaggio semplice e chiaro e di illustrare ogni aspetto particolare dell'argomento con il maggior numero di esempi, compatibilmente con lo spazio che la rivista potrà concedere a questa rubrica. In ogni articolo, inoltre, presenteremo un breve programma o sottoprogramma in linguaggio macchina, spiegandone lo scopo e le tecniche necessarie per la sua realizzazione e messa a punto, utilizzando



sempre SM ed elencandone i comandi passo per passo. Cercheremo inoltre di fornire un panorama il più possibile ampio dei vari metodi che consentono di incorporare i programmi o sottoprogrammi in questione all'interno di un ambiente BASIC.

Generalità

Una grossolana schematizzazione di un qualsiasi calcolatore può portare alla sua descrizione in termini di STRUTTURA della macchina fisica e della sequenza di OPERAZIONI ELEMENTARI che

il calcolatore è in grado di svolgere. Tra i componenti che caratterizzano la struttura del calcolatore possiamo elencare:

I registri. Importanza particolare è rivestita dall'accumulatore, destinato ad ospitare uno dei due operandi di una operazione aritmetica o logica, e dal contatore di programma (in inglese Program Counter, spesso abbreviato in PC), che indica quale istruzione del programma in linguaggio macchina debba essere eseguita.

La memoria. Il suo scopo è duplice, deve contenere istruzioni da eseguire e



ABC del linguaggio macchina

dati necessari alle istruzioni suddette.

L'unità aritmetico-logica. Costituisce uno degli aspetti caratterizzanti il funzionamento dei diversi microprocessori. Deve, comunque, essere generalmente in grado di svolgere, almeno, semplici operazioni aritmetiche, come la somma, la complementazione, ecc. e logiche, come il confronto.

Un'unità di controllo. Il suo compito è di organizzare il lavoro del microprocessore, gestendo lo scambio di informazioni fra i vari registri che compongono il microprocessore e la memoria od i collegamenti con le periferiche connesse al calcolatore.

Spesso ci si riferisce all'insieme della unità aritmetico-logica (Arithmetical-Logical Unit o ALU) e dell'unità di controllo (Control Unit o CU) con il nome di CPU (iniziali di Central Processing Unit, unità centrale di elaborazione).

Le operazioni o istruzioni elementari che il calcolatore è in grado di svolgere coinvolgono, ovviamente, gli oggetti sopra descritti; una istruzione elementare è, ad esempio, rappresentata da

CLA

che azzerava l'accumulatore, o dalla

LDA\$2800

che trasferisce nell'accumulatore il contenuto del byte di memoria il cui indirizzo esadecimale viene specificato dal campo indirizzo dell'istruzione.

Prima di cominciare a parlare di istruzioni in linguaggio ASSEMBLY e di formati di indirizzamento vorremmo però accennare a qualche aspetto fondamentale dei linguaggi di programmazione. In qualsiasi campo dell'attività umana esiste, notoriamente, più di un modo di illustrare un problema. Le varie lingue parlate e scritte del globo non sono che uno dei possibili metodi di comunicazione. Il disegno, la fotografia e, andando verso linguaggi più codificati, la rappresentazione geometrica di un oggetto o di una funzione o la trascrizione di un brano musicale sul pentagramma, rappresentano tutti forme di comunicazione e schematizzazione di un problema o di una sequenza di operazioni.

Nei suoi rapporti con i calcolatori l'uomo si è trovato ad avere a che fare con un non trascurabile problema di collegamento o, come suol dirsi, di interfacciamento tra il linguaggio ed il modo di pensare umani e l'insieme delle istruzioni elementari del calcolatore.

Il linguaggio macchina, l'unico direttamente eseguibile dal calcolatore, è ov-

viamente nato per primo. Per risolvere i grossi problemi di programmazione che un simile linguaggio comportava, sia in termini di tempo di progettazione che di messa a punto, sono nati i vari linguaggi di programmazione con istruzioni più comprensibili all'uomo, o comunque più facili da ricordare, che definiscono le opportune istruzioni da impartire al calcolatore. Questi linguaggi prendono il nome di "linguaggi ad alto livello".

Esempi di linguaggio ad alto livello sono rappresentati dal FORTRAN, dal PASCAL, dal COBOL e, naturalmente, dal BASIC (benché le caratteristiche di questi ultimi siano decisamente meno sofisticate dei linguaggi più evoluti, come i primi due o l'ultimo nato, l'ADA). Nei linguaggi di programmazione ad alto livello le singole istruzioni corrispondono a diverse operazioni elementari del tipo comprensibile da parte del calcolatore e precedentemente descritto. Ad esempio l'istruzione BASIC

ALFA = 0

corrisponde alla ricerca della locazione corrispondente alla variabile ALFA, al caricamento del suo contenuto in accumulatore, al suo azzeramento tramite l'unità aritmetico-logica ed al suo trasferimento alla locazione di partenza.

Dato che, come abbiamo già detto, l'unico linguaggio direttamente eseguibile da parte del calcolatore è rappresentato dalle suddette operazioni elementari, si pone un problema di "conversione" da una forma all'altra di linguaggio di programmazione. Gli utenti del C64 sanno infatti che per poter utilizzare il loro calcolatore è sufficiente accenderlo e scrivere alcune righe di istruzioni BASIC.

Il linguaggio BASIC fornito dal C64 è un linguaggio "interpretato".

Parte della memoria ROM (Read Only Memory, memoria a sola lettura) del calcolatore contiene un programma in linguaggio macchina che si occupa di fare da "interprete" per il calcolatore delle istruzioni BASIC introdotte dall'utente; esattamente come farebbe ciascuno di noi, avendo a che fare con un individuo che parlasse una lingua per noi incomprensibile. Il problema della interpretazione è rappresentato dal fatto che essa avviene passo passo, e non una volta per tutte all'inizio del programma, esattamente come si svolgerebbe la conversazione tra due persone che comunicano attraverso un interprete.

L'"interpretazione" dei linguaggi ad al-

to livello non rappresenta l'unica opzione possibile. Esistono infatti esempi di linguaggi "compilati" (tutti i sistemi medio-grandi presentano, ad esempio, queste caratteristiche). Tanto per conservare il paragone, i compilatori rappresentano una sorta di "traduttori" a cui l'utente consegna il proprio programma in linguaggio ad alto livello perché lo traducano, una volta per tutte, in linguaggio macchina, producendo, dopo un certo tempo, "codice oggetto", che può essere "dato in pasto" al calcolatore perché lo esegua. Qualsiasi modifica venga apportata al programma deve però essere sottoposta da capo all'intera trafila: programmazione in linguaggio ad alto livello, compilazione ed esecuzione.

Un linguaggio simbolico intermedio è l'ASSEMBLY, che, pur non essendo ancora vero e proprio linguaggio macchina (consistendo non di valori numerici, ma di codici mnemonici alfanumerici), si pone in corrispondenza biunivoca con le istruzioni in linguaggio macchina o, per essere più chiari, ha una corrispondenza uno a uno con queste istruzioni. Programmare in linguaggio macchina significa infatti stabilire il contenuto di ogni cella di memoria ed il linguaggio ASSEMBLY consente infatti di operare sul contenuto della memoria e dei vari registri del microprocessore. Anche se non è forse possibile spingere oltre il paragone precedente, potremmo dire che utilizzare il linguaggio ASSEMBLY equivale a cominciare ad avere una conoscenza elementare, ma precisa, delle forme grammaticali e sintattiche di una lingua straniera, come l'infarinatura d'inglese fornita dalla scuola media italiana, che non ci metterà forse in grado di comprendere le sfumature più sottili delle opere di Shakespeare, ma, senz'altro, almeno di chiedere un'informazione ad un passante.

I fondamenti del linguaggio macchina

I problemi che l'apprendimento del linguaggio macchina comporta vengono generalmente, a nostro parere, esagerati. Il nocciolo della questione è rappresentato dalla necessità di apprendere regole fondamentalmente diverse da quelle cui si è da tempo fatta l'abitudine. Il linguaggio ASSEMBLY, infatti, non è particolarmente più complicato del BASIC, anche se senz'altro molto più critico per quanto riguarda la correzione degli errori (ma per questo c'è SMI!). In effetti tutti i pionieri dell'informatica hanno dovuto imparare a programmare

in linguaggio macchina prima che nei vari linguaggi ad alto livello e, cosa piuttosto interessante, alcuni di essi incontrano gli stessi problemi nella programmazione dei linguaggi ad alto livello che l'utente medio deve affrontare per programmare in linguaggio macchina.

Sappiamo, per nostre esperienze, che l'intera questione è rappresentata dalla difficoltà di affrontare un modo "nuovo", diverso dal solito, di programmare; così come tutti coloro che affrontano per la prima volta un calcolatore devono confrontarsi con il diverso modo di pensare e di schematizzare una operazione proprio dei calcolatori.

La prima cosa da apprendere per quanto riguarda la programmazione in linguaggio macchina è il modo in cui i calcolatori "manipolano" l'informazione. Gli antropologi sostengono che la razza umana ha sviluppato un sistema numerico decimale, poiché dispone di dieci dita di cui si è inizialmente servita per contare. Dato che un calcolatore, a causa delle sue limitazioni, non dispone di dieci dita, bensì di una serie di stati elettrici stabili, è stato sviluppato per essi un sistema numerico differente. Nel caso più semplice, in cui gli stati in cui è possibile manipolare e registrare le informazioni sono solo due, il sistema prende il nome di "binario".

Il modo più semplice di descrivere l'aspetto puramente elettrico della numerazione binaria può essere rappresentato dal pensare ogni cella di cui la memoria è composta come collegata ad uno dei due livelli fondamentali di tensione del calcolatore: la "massa" (0) o la tensione di alimentazione (1). Ciò non impedisce che ci possano essere livelli di tensione intermedia per altri scopi pratici.

Un modo ancora più semplice (e non molto originale) di illustrare lo stesso concetto è di pensare ad ogni cifra binaria (in inglese binary digit, contratto in bit) come ad una lampadina, che sia uguale a zero, se spenta od a uno, se accesa. Oppure, sempre pensando a circuiti che operano in condizioni estreme di collegamento, come ad un interruttore a due stati stabili: chiuso o aperto.

Per compiere operazioni aritmetiche o codificare gruppi di segnali è, ovviamente, necessario creare e definire una nuova, anche se in tutto e per tutto simile alla precedente, aritmetica, chiamata, naturalmente, aritmetica binaria.

Nella rappresentazione dei numeri in qualsiasi base — ed in particolare nella consueta base decimale — utilizziamo quella che si è soliti chiamare "notazione posizionale", per fornire una rappresentazione sintetica del numero stesso. In pratica, ciò significa che alla posizione di ciascuna cifra all'interno di un numero è attribuibile un ben preciso "peso", diverso a seconda della posizione e, per lo meno per le cifre arabe, crescente da destra verso sinistra. Il peso in questione è rappresentato da una po-

tenza della base stessa; così il numero 5207, composto da quattro cifre decimali, rappresenta in effetti il valore numerico

$$5 \times 10^3 + 2 \times 10^2 + 0 \times 10^1 + 7 \times 10^0$$

Lo stesso discorso è valido anche per i numeri frazionari:

$$3,201 = 3 \times 10^0 + 2 \times 10^{-1} + 0 \times 10^{-2} + 1 \times 10^{-3}$$

Passando dalla base decimale a qualsiasi altra base B, è necessario ricordare che le cifre di cui si dispone sono quelle da 0 a B-1 (in corrispondenza di B scatta una unità di grandezza immediatamente superiore) e che il peso di ciascuna è dato da una diversa potenza di B in base alla sua posizione.

La più piccola base significativa è ovviamente la base 2, in cui le cifre possono assumere solo i due valori 0 e 1 e le posizioni sono associate alle potenze (positive o negative) di 2. Ad esempio:



Bit 7. 6. 5. 4. 3. 2. 1. 0.
1 0 0 1 0 1 0 1

equivale a

$$1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

cioè

$$128 + 16 + 4 + 1 = 149 \text{ in base decimale}$$

Potreste forse domandarvi come sia possibile rappresentare una base superiore a dieci e se esista un limite superiore a questa rappresentazione. Un limite, ovviamente, non esiste, se non per questioni puramente pratiche. Per quanto riguarda la rappresentazione,

una volta esaurite le cifre arabe di cui ci serviamo abitualmente, è necessario introdurre dei simboli atti ad esprimere i valori compresi tra 9 e B-1 (essendo B la base, maggiore di dieci, prescelta).

Nella base numerica maggiore di dieci più diffusa in informatica (la base 16 della numerazione esadecimale) si è convenuto di utilizzare le prime sei lettere dell'alfabeto per rappresentare le cifre da 10 a 15. Quindi la seguente notazione contenente lettere miste a numeri

1A0F

in base esadecimale significa

$$1 \times 16^3 + 10 \times 16^2 + 0 \times 16^1 + 15 \times 16^0 = 6671$$

La conversione di qualunque numero intero N da una base B ad una base b può essere ottenuto ricorrendo a divisioni ripetute per la base b, operando però in base B. Tutto ciò potrà sembrare molto confuso; in realtà il semplice esempio che segue basterà a spiegarlo. Supponiamo di voler convertire il numero 376 da base decimale a base binaria. Dovremo, innanzi tutto, calcolare il quoziente $376/2$. Il risultato sarà rappresentato da un numero $N1 \geq 0$ (188) con un resto $r0$ per cui vale la seguente relazione

$$0 \leq r0 < b$$

Ripetendo su $N1$ la sequenza di operazioni precedentemente descritte si otterrà un quoziente $N2$ ed un resto $r1$, e così via, fino a che il quoziente non è uguale a zero, ponendo termine al procedimento. Nel nostro caso la sequenza delle operazioni può essere riassunta dalla seguente tabella

N	376	r_i
$N/2 = N1$	188	0
$N1/2 = N2$	94	0
$N2/2 = N3$	47	0
$N3/2 = N4$	23	1
$N4/2 = N5$	11	1
$N5/2 = N6$	5	1
$N6/2 = N7$	2	1
$N7/2 = N8$	1	0
$N8/2 = N9$	0	1

La conversione in cifre binarie del numero decimale 376 può essere ottenuta leggendo ordinatamente dal basso verso l'alto i resti delle singole divisioni; pertanto:

$$376_{10} = 10111100_2$$

dove, con la scrittura precedente, si indica l'equivalenza tra i valori numerici espressi nelle due basi (segnalate dall'indice apposto a ciascun numero). Possiamo verificare il risultato con il metodo della serie di potenze:



ABC del linguaggio macchina

$10111100 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 128 + 32 + 16 + 8 + 4 = 188$

Come è possibile vedere, occorre un numero di cifre maggiore per rappresentare un numero quanto più piccola è la base in cui si intende rappresentarlo. Per il numero del nostro esempio, 188, sono state sufficienti otto cifre binarie. L'esempio non è stato scelto a caso; otto bit, infatti, è la lunghezza normale di una parola di memoria del microprocessore 6510, su cui si basa il C64. Con questo numero di bit è possibile rappresentare tutti i numeri interi compresi tra 0 e 255. In generale, disponendo di d cifre in base B è possibile rappresentare tutti i numeri interi compresi tra

$$0 \leq N \leq B^d - 1$$

Non ci addentreremo, per ora, nei meandri della rappresentazione dei numeri frazionari o della rappresentazione normalizzata in virgola mobile, tramite esponente e mantissa. Affronteremo questi argomenti negli articoli successivi; per il momento occupiamoci delle operazioni in numerazione binaria su numeri interi, utilizzando SM per scrivere e verificare qualche breve programma numerico in linguaggio ASSEMBLY che ci consenta di velocizzare le operazioni di calcolo del nostro C64.

Operazioni su dati binari

Le regole aritmetiche sui numeri binari non differiscono sostanzialmente da quelle per dati decimali e sono immediatamente ricavabili da esse. Sommando due bit otterremo:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \\ 1 + 0 &= 1 \\ 1 + 1 &= (1) 0 \end{aligned}$$

dove (1) indica, ovviamente, un riporto (in inglese "carry") sulla cifra binaria più significativa (la scrittura 10, in base due, equivale infatti al numero decimale "2").

Utilizziamo ora il nostro monitor per scrivere e provare un brevissimo programma in linguaggio ASSEMBLY che realizza la somma di due numeri a otto bit (quindi, come abbiamo visto, interi e minori o uguali a 255) in linguaggio macchina. Prenderemo confidenza, contemporaneamente, con le opzioni offerte da SM.

Per poter realizzare il programma prefissato dobbiamo però conoscere un po'

meglio il funzionamento dell'ALU del microprocessore 6510. L'unità aritmetico-logica è caratterizzata dal fatto di disporre di due ingressi, contrariamente alla maggior parte dei registri che dispongono di un solo ingresso tramite cui comunicare con il resto del calcolatore. La funzione dell'ALU consiste, ovviamente, nell'esecuzione di determinate operazioni aritmetiche e logiche sui dati che compaiono alle sue due porte di ingresso. Uno dei due ingressi dell'ALU è sempre rappresentato dall'accumulatore, quindi nelle operazioni aritmetico-logiche l'ALU assumerà automaticamente come uno dei due operandi il contenuto dell'accumulatore, mentre un eventuale (esistono infatti delle operazioni, che assumono il nome di "unarie" - come la complementazione o cambiamento di segno -, che non prevedono un secondo operando) secondo operando si troverà in una locazione di memoria, il cui indirizzo verrà specificato dall'istruzione stessa, che chiama in causa l'unità aritmetico-logica.

Il risultato di qualsiasi operazione che coinvolga l'ALU verrà depositato nell'accumulatore. Il suo utilizzo comporta un notevole risparmio di tempo, in quanto uno dei due ingressi non deve venire specificato, ma viene assunto automaticamente o, come suol dirsi, per "default". Ciò comporta però lo svantaggio di dover costantemente caricare e scaricare dati per e dall'accumulatore, prima e dopo ogni operazione. Per risolvere il nostro semplice problema potremmo ricorrere allo schema seguente:

- Caricare in accumulatore il contenuto della cella di indirizzo IND1
- Sommare al contenuto dell'accumulatore (tramite la ALU) il contenuto della cella di memoria di indirizzo IND2
- Trasferire il risultato, contenuto nell'accumulatore, nella cella di memoria di indirizzo IND3 (supponendo di dover utilizzare il contenuto di IND1 e IND2 per successive operazioni).

Questa sequenza di istruzioni prevede, ovviamente, che ogni volta che desideriamo operare la somma di due numeri di otto bit il primo venga posto, in qualcuno dei modi che vedremo, in corrispondenza dell'indirizzo IND1 ed il secondo in IND2, mentre il risultato venga letto in IND3.

I lettori più attenti avranno anche notato che il nostro programma non darà risultati in ogni caso corretti; infatti, se la somma dei due operandi è superiore

a 255, il risultato non può essere contenuto in soli otto bit, ma necessita di un ulteriore bit, di riporto. A questo scopo all'ALU è collegato un registro, nella letteratura del 6510 generalmente indicato con P e chiamato "registro dei flag di stato" (status flag register). Negli Stati Uniti prendono il nome di "flag" le asticcioline metalliche a forma di bandierina (flag significa appunto bandiera, in inglese) sulle cassette postali (ben note per la loro forma grazie ai cartoni animati ed alle strisce di "Peanuts"), che "segnalano" la presenza di pacchi o lettere da ritirare. Lo scopo del registro dei flag di stato è appunto quello di un semaforo, che consenta di rilevare o segnalare situazioni particolari dello stato del microprocessore. Vedremo di volta in volta, quando si presenterà l'occasione, il significato di ciascuno degli otto bit del registro P. Per il momento è sufficiente dire che ciascuno di essi è fisicamente realizzato tramite un "flip-flop" o "bistabile", un dispositivo cioè con due soli stati del quale, inviando un impulso, è possibile far "commutare" lo stato, vale a dire "abbassare" la bandierina, se questa è "alzata" o "alzarla", se è sollevata. Questo particolare registro ci interessa nel nostro caso, in quanto il suo bit meno significativo (quello cioè di "minor peso"), nella posizione più a destra, è il bit che segnala il riporto di una operazione dell'ALU. È quindi in questo registro, analizzando il bit 0, che dovremo cercare l'eventuale riporto della nostra operazione di somma. Per il momento, però, trascureremo di considerare anche il riporto, e la necessità di azzerarne il valore prima di ogni operazione, per concentrarci sul meccanismo della nostra semplice operazione.

Carichiamo, quindi, il programma apparso sullo scorso numero di "SUPER-COMMODORE" ed attiviamolo tramite una istruzione SYS49152. Detto per inciso, questa istruzione manda in esecuzione il programma in linguaggio macchina che abbiamo caricato da nastro o da disco direttamente in memoria e che inizia in corrispondenza della locazione 49152.

Vedremo comparire le righe seguenti:

```
B* PC  IRQ  SR  AC  XR  YR  SP
.; C04E C351 32 20 00 00 F6
```

Nella prima riga vengono elencati i registri del microprocessore e nella seconda, al di sotto di ognuno, il contenuto attuale. Da sinistra a destra essi sono: il "Program Counter" (che indica l'istruzione corrente), il vettore delle interruzioni (il cui significato affronteremo in

seguito), il registro dei flag di stato ("Status Register"), l'accumulatore, i registri ausiliari X e Y ed il puntatore alla pila ("Stack Pointer"); affronteremo dettagliatamente ciascuno degli ultimi tre quando se ne presenterà l'occasione.

Il contenuto di ognuno è indicato, come probabilmente avrete intuito data la presenza di lettere miste a numeri, in formato esadecimale. Nello stesso formato dovremo esprimere tutti gli indirizzi utilizzati dalle nostre istruzioni Assembly. Si tratta, a questo punto, di stabilire quali locazioni utilizzare per immagazzinare gli operandi ed il risultato del nostro semplice programma, oltre al linguaggio macchina corrispondente alle istruzioni che lo compongono. Con 64K byte a disposizione, naturalmente, non c'è che l'imbarazzo della scelta.

Dovremo però fare attenzione a non interferire con eventuali programmi BASIC che utilizzino il nostro sottoprogramma in linguaggio macchina o con locazioni utilizzate dal calcolatore per scopi particolari (ad esempio, il controllo della grafica o degli effetti sonori).

Per brevi sottoprogrammi in linguaggio macchina come il nostro, un gruppo di locazioni molto comode è rappresentato dal buffer del secondo registratore collegabile al calcolatore (nell'ipotesi, naturalmente, di non servirsene).

Questo gruppo di locazioni è compreso tra gli indirizzi decimali 828 e 1019, ed offre il vantaggio di non richiedere alcun tipo di operazione che coinvolga i puntatori alla memoria riservata al BASIC, per proteggere il programma in linguaggio ASSEMBLY dall'interferenza di eventuali programmi BASIC.

Diciamo, quindi, che IND1 corrisponderà a 828, IND2 a 829, IND3 a 830 e che il codice macchina, che costituirà il nostro programma, sarà ospitato a partire dalla locazione 831 e richiamabile quindi con una istruzione SYS831.

Come detto, però, a noi interessa conoscere gli indirizzi esadecimali corrispondenti alle locazioni suddette. Un sistema consiste, naturalmente, nel calcolarsi con pazienza l'equivalente di ciascun valore desiderato tramite il metodo sopra descritto. Un sistema più comodo è rappresentato da una delle opzioni di SM. Dopo il punto che compare sullo schermo digitiamo

828

— il simbolo '#' contraddistingue i valori numerici decimali, così come '\$' gli esadecimali — e premiamo RETURN; alla destra dei caratteri da noi inseriti vedremo apparire, da sinistra a destra, la conversione esadecimale del numero decimale indicato, i due caratteri ASCII la cui codifica corrisponde ai due byte che lo compongono e la sua conversione binaria (sempre su sedici bit, cioè due byte).

Una delle caratteristiche del linguaggio ASSEMBLY è rappresentata dal fatto

che l'indirizzo della maggior parte delle istruzioni prevede l'impiego di due byte (quattro cifre esadecimali, cioè) e che il meno significativo dei due vada sempre premessa alla più significativa.

Pertanto, per implementare la prima operazione del nostro programma, cioè il caricamento in accumulatore del contenuto della cella di indirizzo decimale 828 (esadecimale 033C), dovremo inserire, dopo il codice operativo corrispondente all'istruzione "Carica in accumulatore" (codice mnemonico Assembly corrispondente LDA, cioè Load in Accumulator, appunto), la codifica esadecimale dell'indirizzo corrispondente ad IND1, preceduta dal simbolo \$ per indicare la base utilizzata.

Come detto, però, il calcolatore non ospita istruzioni in codice mnemonico, ma la corrispondente codifica sotto forma di coppie di cifre esadecimali. Anche in questo caso esistono due strade possibili. La più noiosa consiste nell'armarsi di molta pazienza e scartabellare



uno dei dizionari Assembly, che i volumi citati nell'articolo precedente mettono generalmente a disposizione, convertendo manualmente ogni istruzione in codice mnemonico in un numero composto da due cifre esadecimali. I problemi non si fermano qui, però. La maggior parte delle istruzioni in codice mnemonico, infatti, presenta più di una codifica (spesso quattro-cinque, fino a otto diverse) in base al tipo di indirizzamento, un argomento che affronteremo sen'altro nel prossimo articolo.

La strada più semplice consiste quindi nel far svolgere il lavoro di conversione ad SM. Per farlo, di seguito al punto che appare all'inizio della riga, è necessario digitare la lettera A ("Assembler") seguita dalla codifica esadecimale dell'indirizzo a partire da cui si desidera inserire codice macchina. Dato che abbiamo deciso di far seguire il "testo" del nostro programma ASSEMBLY agli in-

dirizzi che ospitano gli operandi ed il risultato, nel nostro caso l'indirizzo suddetto sarà 033F (non è necessario specificare che la base è esadecimale, tramite il simbolo '\$', dal momento che l'indirizzo viene considerato esadecimale "per default").

Di seguito digiteremo le tre lettere dell'istruzione in codice mnemonico (nel nostro caso LDA) seguite dall'indirizzo del primo operando (\$033C). Lo schermo avrà il seguente aspetto

.A 033F LDA \$033C

Assicuriamoci di non aver commesso errori e premiamo RETURN. Lo schermo si trasformerà come segue

.A 033F AD 3C 03 LDA \$033C

inserendo tra l'indirizzo iniziale e l'istruzione in codice mnemonico la codifica esadecimale di quest'ultima, seguita dall'indirizzo dell'operando nel formato "byte - meno - significativo" / "byte - più - significativo".

Sulla riga successiva, dopo il solito punto, apparirà la lettera A seguita dall'indirizzo immediatamente successivo. Dato che la prima istruzione ed il relativo indirizzo occupano tre byte, otterremo

.A 0342

Possiamo ora inserire la nostra seconda istruzione (aggiungi all'accumulatore il contenuto di IND2), la cui trascrizione in codice mnemonico è

.A 0342 ADC \$033D

dove ADC (Add to accumulator with Carry, cioè addizione con riporto) rappresenta la codifica dell'operazione voluta e \$033D l'indirizzo esadecimale corrispondente a IND2. Premendo RETURN il calcolatore inserirà la conversione in cifre esadecimali dell'istruzione inserita

.A 0342 6D 3D 03 ADC \$033D

L'ultima istruzione della nostra breve "routine" in linguaggio macchina avrà lo scopo di trasferire il risultato dell'operazione precedente dall'accumulatore alla locazione corrispondente ad IND3. In codice macchina questa operazione viene svolta dall'istruzione STA (Store Accumulator in memory, registra il contenuto dell'accumulatore all'indirizzo di memoria specificato). Sulla riga corrente di schermo comparirà quindi

.A 0345

e noi aggiungeremo

.A 0345 STA \$033E

Tra l'indirizzo dell'istruzione corrente ed il codice mnemonico il calcolatore



ABC del linguaggio macchina

inserirà la trascodifica dell'istruzione

.A 0345 8D 3E 03 STA \$033E

Il nostro programma può ritenersi, a questo punto, concluso. Per poterlo richiamare dall'interno di un programma BASIC, però, è necessario aggiungere un'ultima istruzione, equivalente alla istruzione BASIC RETURN, che deve chiudere ogni chiamata a sottoprogramma (GOSUB) per un corretto svolgimento del programma BASIC. L'istruzione in questione si chiama semplicemente RTS (ReTurn from Subroutine, rientro da sottoprogramma) e provoca il rientro in ambiente BASIC.

Sulla riga corrente

.A 0348

digiteremo quindi

.A 0348 RTS

ed SM aggiungerà

.A 0348 60 RTS

Come potete notare, non è necessario specificare alcun argomento per l'istruzione, dal momento che l'indirizzo di rientro del sottoprogramma è contenuto nella pila (Stack) il cui elemento superiore è individuato dal registro apposito (SP, cioè Stack Pointer, puntatore alla pila).

Il monitor vi offrirà di inserire altre istruzioni nella locazione immediatamente successiva

.A 0349

Premendo RETURN si uscirà dalla funzione Assembler e si rientrerà sotto il "Prompt"

Premuriamoci, innanzi tutto, di registrare una copia del nostro sottoprogramma. Per farlo è necessario specificare, oltre al nome ed alla periferica, l'indirizzo di partenza e l'indirizzo immediatamente seguente a quello conclusivo (vale a dire l'indirizzo che compare sulla riga di Assembler al momento dell'uscita dalla funzione). Utilizzeremo quindi il seguente comando

.S 033F 0349 "ADDIZ. A 8 BIT" 08

dove 08 individua come periferica l'unità dischi (tralasciando il numero della periferica, o specificando 01, si utilizze-

rà l'unità nastro). Dopo che il calcolatore avrà portato a termine la registrazione potremo chiedere di vedere la Directory del disco (nel caso in cui avessimo utilizzato questo tipo di registrazione), premendo

>

seguito da RETURN.

Possiamo ora "provare" il funzionamento del nostro breve sottoprogramma. Per farlo, naturalmente, è necessario innanzi tutto predisporre il contenuto delle locazioni 828 e 829, destinate ad ospitare i due operandi della nostra operazione. Non è necessario uscire da SM per veder funzionare il nostro programma ed il monitor ci permetterà inoltre di disporre di alcune funzioni che, soprattutto per lo sviluppo di programmi più complicati, si riveleranno preziose.

Cominciamo con l'utilizzare la funzione Memory display, in modo da "vedere" le locazioni in questione. Per farlo è sufficiente digitare il comando M seguito dall'indirizzo esadecimale della prima locazione che si desidera esaminare. Il calcolatore mostrerà anche il contenuto delle sette locazioni seguenti

.M 033C C0 4E 32 AD 3C 03 6D 3D

Nell'istruzione precedente 033C corrisponde alla codifica esadecimale dell'indirizzo della locazione 828, il cui contenuto attuale è pari a C0 in esadecimale (cioè $12 \times 16 + 0 \times 1 = 192$ in decimale). La stessa funzione ci permette di vedere che la locazione 829 (corrispondente ad IND2) "vale" attualmente 4E e la 830, sempre in esadecimale, 32.

Il quarto byte mostrato dalla stessa funzione corrisponde all'indirizzo 831, a partire dal quale abbiamo immagazzinato il "testo" del nostro breve sottoprogramma; infatti, la codifica della prima istruzione e di parte della seconda è contenuta negli ultimi cinque byte mostrati dalla riga precedente.

La stessa funzione ci consente anche di modificare il contenuto delle locazioni. Spostiamo il cursore sulla riga precedente, tramite i tasti di controllo del cursore CRSR (contrassegnati dalle frecce e posti alla destra della riga inferiore della tastiera), e modifichiamo i due primi byte come segue

.M 033C AA 1B 32 AD 3C 03 6D 3D

Per far sì che i due numeri esadecimali che abbiamo sovrapposto a quelli contenuti nelle due locazioni suddette vengano effettivamente inseriti in memoria

è necessario premere RETURN. Il monitor vi offrirà di visualizzare gli otto byte successivi, mostrando, sulla riga seguente, l'istruzione

..33C

È sufficiente premere RETURN per uscire dalla funzione e vedersi riproporre il consueto "prompt".

Potreste chiedervi qual è il valore delle due cifre esadecimali inserite nelle locazioni di memoria destinate a contenere gli operandi. SM dispone di una comoda funzione di conversione esadecimale (analoga a quella già vista per la conversione decimale). Per attivarla è sufficiente inserire le quattro cifre esadecimali che si desidera convertire precedute dal simbolo '\$'. Alla pressione del tasto RETURN compariranno sulla stessa riga, a fianco dei caratteri digitati, la conversione decimale, i simboli ASCII corrispondenti e la codifica binaria del numero esadecimale desiderato.

Ad esempio, battendo

.\$ 00AA

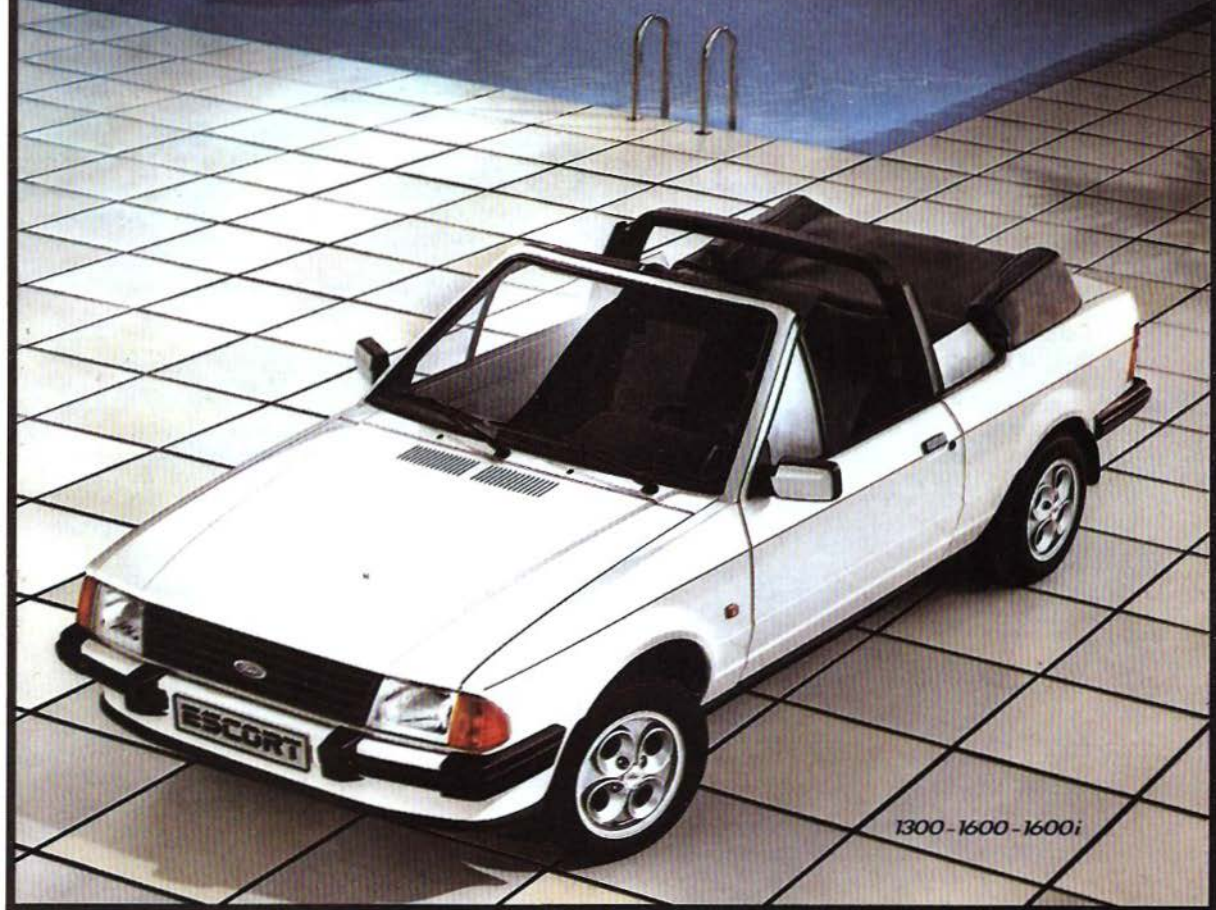
si otterrà il valore decimale corrispondente, pari a $10 \times 16 + 10 = 170$.

N.B. - SM accetta solo numeri esadecimali di quattro cifre: per convertire numeri aventi un numero di cifre inferiore è necessario premettere degli zeri non significativi fino a raggiungere quattro cifre. Non è necessario convertire numeri esadecimali di più di quattro cifre, in quanto nessun indirizzo del microprocessore supera i quattro byte di lunghezza e le sue istruzioni non ammettono operandi di più di due byte.



(Continua sul prossimo numero)

CONVERTIBLE BY FORD.



1300 - 1600 - 1600i

Un modo di essere. Il piacere di toccare il mondo. Un desiderio che diventa automobile.

Escort Cabriolet. Quattro comodi posti, un vero bagagliaio che offre una capienza ancora superiore, grazie alla possibilità di reclinare i sedili posteriori.

Escort Cabriolet è soluzioni tecnico-costruttive d'avanguardia: roll-bar integrale, capote completamente foderata e insonorizzata, ripiegabile facilmente senza il minimo sforzo. Lunotto posteriore in vetro riscaldato elettricamente.

Escort Cabriolet è straordinarie prestazioni. Come nella versione 1.6 injection: 186 km/h, da 0 a 100 km/h in 9.9 sec.

Escort Cabriolet. Il piacere dell'automobile allo stato puro.

Prestazioni e consumi	1296 cc	1597 cc	1597 cc-i
Velocità massima	157 km/h	169 km/h	186 km/h
Da 0 a 100 km/h	14.0 sec	11.8 sec	9.9 sec
Consumo a 90 km/h	17.8 km/lt	18.2 km/lt	16.7 km/lt

Tutte le vetture Ford sono coperte da garanzia 1-3-6 (un anno di garanzia estensibile a tre con "La Lunga Protezione" e sei anni di garanzia contro la corrosione perforante) e assistite in oltre 1000 punti di servizio. Finanziamenti Ford Credit e cessioni in Leasing.

ESCORT

Tecnologia e temperamento.



Piccola guida per l'input dei programmi

Molti dei programmi listati da **SUPERCOMMODORE** contengono particolari caratteri di controllo (controllo cursore, tasti colore, video-inverse, ecc.). Per rendere chiaro ciò che dev'essere battuto quando è necessario inserire uno dei suddetti caratteri sono state stabilite le seguenti convenzioni.

Generalmente i listati per VIC 20 e C64 contengono delle «parole» racchiuse tra parentesi graffe { }; tali parole rappresentano particolari caratteri di controllo: **{GIU}** significa premere il simbolo del cursore verso il basso. **{5 SPAZI}** vuol dire battere 5 volte la barra-spazio.

Per indicare che un tasto dev'essere «shiftato», cioè premuto insieme al tasto SHIFT, nel listato il simbolo di questo tasto sarà sottolineato.

Per esempio S significa che dev'essere premuto il tasto S mentre è abbassato il tasto SHIFT.

La manovra farà apparire sullo schermo un piccolo «cuore».

Trovando un simbolo sottolineato chiuso tra parentesi graffe (esempio **{10 N}**) dovrà essere interpretato come «premere il carattere indicato per il numero di volte che lo precede nelle parentesi»: nel nostro esempio premere 10 volte il tasto N «shiftato».

Se il tasto da premere è indicato tra le parentesi [()], vuol dire che lo stesso dovrà essere premuto mentre è premuto il tasto «Commodore» (il tasto «Commodore» è quello posto nell'angolo in basso a sinistra).

Se il tasto scritto tra [()] è preceduto da un numero, ciò significa che il tasto dev'essere premuto per il numero di volte indicato.

Raramente si potrà trovare un carattere alfabetico solitario racchiuso tra parentesi graffe. Tale carattere sul C64 può essere battuto mentre è premuto il tasto CTRL.

Ad esempio **{A}** sta ad indicare la sequenza CTRL-A. A proposito del «modo virgolette» è nota la possibilità di muovere il cursore sullo schermo con i tasti CRSR. Ogni tanto i programmatori desiderano muovere il cursore durante l'esecuzione del programma. È per questo motivo che nei programmi si troveranno dei **{SIN}**, **{HOME}** e **{BLU}**.

L'unico modo perché il calcolatore distingua il comando diretto da quello inserito nel programma è il «modo virgolette». Premendo il tasto «virgolette» (tasti SHIFT 2) il calcolatore si predispone appunto in «modo virgolette».

Battendo un carattere qualsiasi, cercando poi di correggerlo muovendo il cursore, si otterrà solo un tratto verticale in un quadratino inverso. Questo è infatti il simbolo del cursore a sinistra nel «modo virgolette». L'unico comando di editing non utilizzabile all'interno di un programma è il DEL. Battendo nuovamente le «virgolette» il calcolatore lascia il «modo virgolette». Si può accedere al «modo virgolette» quando inserite spazi nella linea.

In ogni caso, il sistema più semplice per uscire dal «modo virgolette» è quello di premere il tasto RETURN.

Utilizzare la tabella che segue quando è necessario inserire comandi relativi al cursore e ai tasti colore.

Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:	Quando leggete:	Premete:	Vedrete:
{CLR}	SHIFT CLR/HOME		{CYN}	CTRL 4		[<7>]	G 7	
{HOME}	CLR/HOME		{PUR}	CTRL 5		[<8>]	G 8	
{SU}	SHIFT ↑ CRSR		{GRN}	CTRL 6		{F1}	F1	
{GIU}	↓ CRSR		{BLU}	CTRL 7		{F2}	F2	
{SIN}	SHIFT ⇐ CRSR		{YEL}	CTRL 8		{F3}	F3	
{DES}	⇐ CRSR		[<1>]	G 1		{F4}	F4	
{RVS}	CTRL 9		[<2>]	G 2		{F5}	F5	
{OFF}	CTRL 0		[<3>]	G 3		{F6}	F6	
{BLK}	CTRL 1		[<4>]	G 4		{F7}	F7	
{WHT}	CTRL 2		[<5>]	G 5		{F8}	F8	
{RED}	CTRL 3		[<6>]	G 6				

Il tuo settimanale



Il primo settimanale di software su carta per il tuo personal computer. Un appuntamento al quale non puoi mancare ogni venerdì in edicola. Nato in edizione unica, per i più diffusi home e personal computer, dal 1 Aprile, allo stesso prezzo, saranno in edicola le versioni dedicate ai prodotti Sinclair (ZX81, ZX Spectrum e QL) e ai prodotti Commodore (VIC20, C16 e C64).

In più un'edizione esclusiva per gli utenti dei personal Apple, gli home computer Texas e i nuovissimi sistemi MSX. Una grossa novità: la possibilità di abbonarsi dal 1 Aprile 1985 al 31 Dicembre 1985 per ricevere 38 numeri del settimanale comodamente a casa con un notevole risparmio. Chi si abbona riceverà ogni 15 giorni 2 numeri di PAPERSoft

in anticipo rispetto all'uscita in edicola. Non perdere questa occasione!

Compila subito il coupon qui riportato e spediscilo a:

**J.soft
V.le Restelli, 5
20124 MILANO**



EDITRICE  **J.soft**

Tagliando abbonamento a PaperSoft da inviare in busta chiusa a:
J.soft - V.le Restelli, 5 - 20124 MILANO

- ☐ Abbonamento a 38 numeri di PaperSoft al prezzo speciale di L. 30.000 anziché L. 38.000
☐ edizione Sinclair (PS01) ☐ edizione Commodore (PC01) ☐ edizione Apple/T199/MSX (PV01)
☐ contanti allegati ☐ assegno allegato n. _____
☐ ho versato l'importo sul c.c.p. n. 19445204 intestato a J.soft - MILANO

Cognome _____ Nome _____
Via _____ Città _____
CAP _____ Prov. _____ Data _____
Firma _____

MLX per VIC 20 e C64

(versione 2.0 per C64)

di C. Brannon
trad. e adatt.
di M. Cristuib Grizzi
e F. Stella

M LX è un programma che permette di inserire listati in linguaggio macchina esenti in modo assoluto da errori e senza la perdita di tempo del dover battere e controllare lunghe sequenze di istruzioni DATA.

Molti dei nostri listati di programmi in linguaggio macchina hanno il formato MLX (li riconoscete dal fatto che sono esclusivamente numerici) e richiedono quindi che MLX sia caricato in memoria ed eseguito prima della battitura del listato.

MLX vi chiederà l'indirizzo della locazione di partenza e quello della locazione finale del programma da caricare. Questi valori sono sempre indicati nell'articolo che accompagna il listato. MLX vi segnala automaticamente gli errori di battitura MENTRE STATE DIGITANDO IL LISTATO e vi chiede di reinserire la linea errata. L'ultimo numero battuto di ogni

linea rappresenta il checksum e viene visualizzato in reverse.

Sono inoltre disponibili altri comandi, quali: SHIFT-N nuovo indirizzo: permette di cambiare l'indirizzo della linea che volete battere ed è utile nel caso si inseriscano i listati in più parti.

SHIFT-D display: lista i dati inseriti tra due indirizzi di inizio e fine.

SHIFT-L load: carica su nastro o disco un file in formato MLX.

Una volta battuto il listato, e salvatolo tramite MLX, si potrà caricare direttamente il programma con un'istruzione LOAD "nome del file",1,1 per il registratore, oppure LOAD "nome del file",8,1 per l'unità a dischi, salvo diverse indicazioni date negli articoli corrispondenti ai vari programmi. Il programma sarà generalmente mandato in esecuzione con una SYS (indirizzo di partenza).

La versione originaria di MLX per C64 ha subito, dalla prima pubblicazione, diversi ritocchi e miglioramenti, fino alla versione 2.0 qui presentata.

È stato in primo luogo modificato il colore di fondo dello schermo e del bordo, in modo da risultare meno stancante alla vista; sono state ulteriormente perfezionate le routine in linguaggio macchina di salvataggio e caricamento dei programmi; è cambiata la forma del cursore e, cosa più importante, è stato aggiunto un tastierino numerico per gli utenti che si trovano più a loro agio con una diversa disposizione dei tasti.

Oltre a poter usare i soliti tasti numerici, sono stati ridefiniti alcuni tasti alfabetici, in modo da generare ugualmente dei numeri. I tasti ridefiniti sono i seguenti:

U I O	7 8 9
H J K L	diventano 0 4 5 6
M, .	1 2 3

Le persone abituate ad usare tali tastierini accresceranno senz'altro, in brevissimo tempo, la velocità di battitura dei programmi in linguaggio macchina.

MLX versione per VIC 20

```

100 PRINT "{CLR} {PUR}"; CHR$(142); CHR$(8);
                                     :rem 181
101 POKE788,194:REM DISABILITA RUN/STOP
                                     :rem 144
110 PRINT "{RVS}{ 14 SPAZI}"
                                     :rem 117
120 PRINT "{RVS}{DES}{OFF}{<*>}{RVS}
{DES}{DES}{ 2 SPAZI}{<*>}{OFF}{<*>}
{RVS}{RVS}"
                                     :rem 191
130 PRINT "{RVS}{DES}{<G>}{DES}
{ 2 DES}{OFF}{RVS}{<*>}{OFF}{<*>}
{RVS}{OFF}"
                                     :rem 122
140 PRINT "{RVS}{ 14 SPAZI}"
                                     :rem 120
200 PRINT "{ 2 GIU' } {PUR}{BLK} UN PROGRAMMA
PER L'IN-TRODUZIONE{ 2 SPAZI} DI ROUT
INEIN LINGUAGGIO MACCHINA";
                                     :rem 202
205 PRINT "A PROVA DI ERRORE{ 5 GIU' }"
                                     :rem 226
210 PRINT "{BLK}{ 4 SU} INDIRIZZO DI PARTEN
ZA": INPUTS: F=1-F: C$=CHR$(31+119*F)
                                     :rem 52
220 IFS<256ORS>32767 THEN GOSUB3000:GOTO210
                                     :rem 2
225 PRINT:PRINT:PRINT:PRINT
                                     :rem 123
230 PRINT "{BLK}{ 3 SU} INDIRIZZO CONCLUSIV
O": INPUTS: F=1-F: C$=CHR$(31+119*F)
                                     :rem 170
240 IFE<256ORE>32767 THEN GOSUB3000:GOTO230
                                     :rem 234
250 IFE<STHENPRINTC$; "{RVS} INDIRIZZO CONC
LUSIVO <INDIRIZZO DI PARTENZA
{ 3 GIU' }": GOSUB1000:GOTO230:rem 27
260 PRINT:PRINT:PRINT
                                     :rem 179
300 PRINT "{CLR}"; CHR$(14):AD=S
                                     :rem 56
310 PRINTRIGHT$ ("0000"+MID$(STR$(AD),2),5
);":":FORJ=1TO6
                                     :rem 234
320 GOSUB570:IFN=-1 THEN J=J+N:GOTO320
                                     :rem 228
390 IFN=-211 THEN 710
                                     :rem 62
400 IFN=-204 THEN 790
                                     :rem 64
410 IFN=-206 THEN PRINT:PRINT "{GIU' } INSERIS
CI UN NUOVO{ 4 SPAZI} INDIRIZZO"; INPU
TZZ
                                     :rem 42

```

```

415 IFN=-206 THEN IFZZ<SORZZ> E THEN PRINT"
{RVS} ESCE DAL CAMPO DI VA-" :rem 150
416 IFN=-206 THEN IFZZ<SORZZ> E THEN PRINT"
{RVS} LORI INDICATO": GOSUB1000:GOTO410
                                     :rem 105
417 IFN=-206 THEN AD=ZZ:PRINT:GOTO310
                                     :rem 238
420 IFN<>-196 THEN 480
                                     :rem 133
430 PRINT: INPUT "LISTATO: DA"; F: PRINT, "
{ 2 SIN} A"; INPUT T
                                     :rem 29
440 IFF<SORE> E ORT<SORT> E THEN PRINT "MINIMO"
;S; "{SIN}, MASSIMO"; E:GOTO430
                                     :rem 36
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$ ("000
0"+MID$(STR$(I),2),5);":":
                                     :rem 30
455 FORK=0TO5:N=PEEK(I+K):IFK=3 THEN PRINTS
PC(10);
                                     :rem 34
457 PRINTRIGHT$ ("00"+MID$(STR$(N),2),3);"
,";
                                     :rem 157
460 GETA$: IFA$>" THEN PRINT:PRINT:GOTO310
                                     :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
T:GOTO310
                                     :rem 50
480 IFN<0 THEN PRINT:GOTO310
                                     :rem 168
490 A(J)=N:NEXTJ
                                     :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
SUM=(CKSUM+A(I))AND255:NEXT
                                     :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20)
                                     :rem 234
515 IFN=CKSUM THEN 530
                                     :rem 255
520 PRINT:PRINT "LA RIGA E' STATA INSE-RIT
A IN MANIERA ERRATA"
                                     :rem 39
525 PRINT "RI-INSERIRLA":PRINT:GOSUB1000:G
OTO310
                                     :rem 167
530 GOSUB2000
                                     :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT
                                     :rem 80
550 AD=AD+6:IFAD<ETHEN310
                                     :rem 212
560 GOTO710
                                     :rem 108
570 N=0:Z=0
                                     :rem 88

```

```

580 PRINT"[<+>]"; :rem 79
581 GETAS:IFA$=""THEN581 :rem 95
585 PRINTCHR$(20);A=ASC(A$):IFA=13ORA=44 :rem 229
ORA=32THEN670 :rem 137
590 IFA>128THENN=-A:RETURN :rem 10
600 IFA<>20THEN630 :rem 172
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT" :rem 109
{SIN}{SIN}";:GOTO690 :rem 105
620 GOTO570 :rem 106
630 IFA<48ORA>57THEN580 :rem 229
640 PRINTA$;N=N*10+A-48 :rem 71
650 IFN>255THENA=20:GOSUB1000:GOTO600 :rem 114
:rem 240
660 Z=Z+1:IFZ<3THEN580 :rem 149
670 IFZ=0THENGOSUB1000:GOTO570 :rem 68
680 PRINT",";:RETURN :rem 205
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211) :rem 7
:rem 236
692 FORI=1TO3:T=PEEK(S%-I) :rem 199
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT :rem 128
:rem 30
700 PRINTLEFT$("{ 3 SIN",I-1);:RETURN :rem 158
:rem 3
710 PRINT"{CLR}{RVS}*** SAVE *** :rem 109
{ 3 GIU' }" :rem 69
720 INPUT"{GIU' } _NOME DEL FILE";F$ :rem 12
730 PRINT:PRINT"{ 2 GIU' }{RVS}N{OFF}ASTRO :rem 124
O {RVS}D{OFF}ISCO: (N/D)" :rem 111
740 GETAS:IFA$<>"N"ANDAS<>"D"THEN740 :rem 126
:rem 85
750 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$ :rem 62
:rem 64
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$) :rem 64
):POKE782,ZK/256 :rem 119
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 117
T$):SYS65469 :rem 130
763 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 130
66 :rem 22
765 POKE254,S/256:POKE253,S-PEEK(254)*256 :rem 119
:POKE780,253 :rem 119
766 POKE782,E/256:POKE781,E-PEEK(782)*256 :rem 119
:SYS65496 :rem 124
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780 :rem 111
:rem 111

```

MLX versione 2.0 per C64

```

100 PRINT"{CLR}{CYN}";CHR$(142);CHR$(8);: :rem 71
POKE53280,0:POKE53281,0 :rem 89
101 POKE788,52:REM DISABILITA RUN/STOP :rem 176
:rem 176
110 PRINT"{RVS}{ 40 SPAZI}"; :rem 220
120 PRINT"{RVS}{ 15 SPAZI}{DES}{OFF}{<*>} :rem 120
~{RVS}{DES}{DES}{ 2 SPAZI}{<*>}{OFF} :rem 250
[<*>]~{RVS}~{RVS}{ 13 SPAZI}"; :rem 220
:rem 220
130 PRINT"{RVS}{ 15 SPAZI}{DES}{<N>}{<H>} :rem 120
{DES}{ 2 DES}{OFF}~{RVS}~{<*>}{OFF} :rem 120
[<*>]{RVS}{ 13 SPAZI}"; :rem 120
140 PRINT"{RVS}{ 40 SPAZI}"; :rem 120
200 PRINT"{ 2 GIU' }{PUR}EDITOR DI CODICE :rem 126
MACCHINA VERSIONE 2.0{ 3 GIU' }" :rem 85
:rem 85
210 PRINT"[<5>]{ 2 SU} INDIRIZZO DI PARTE :rem 228
NZA{ 2 SPAZI}";:INPUTS:F=1-F:C$=CHR$( :rem 62
31+119*F) :rem 64
220 IFS<256OR(S>40960ANDS<49152)ORS>53247 :rem 13
THENGOSUB3000:GOTO210 :rem 119
225 PRINT:PRINT:PRINT :rem 105

```

```

775 PRINT"{GIU' }OPERAZIONE CONCLUSA.":END :rem 152
:rem 8
780 PRINT"{GIU' }ERRORE DI REGISTRAZIONE. :rem 103
RIPROVA.":IFDV=1THEN720 :rem 115
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#15 :rem 212
;E2$:CLOSE15:GOTO720 :rem 215
782 GOTO720 :rem 127
790 PRINT"{CLR}{RVS}*** LOAD *** :rem 820
{ 2 GIU' }" :rem 28
800 INPUT"{ 2 GIU' } _NOME DEL FILE";F$ :rem 157
:rem 2
810 PRINT:PRINT"{ 2 GIU' }{RVS}N{OFF}ASTRO :rem 107
O {RVS}D{OFF}ISCO: (N/D)" :rem 654
820 GETAS:IFA$<>"N"ANDAS<>"D"THEN820 :rem 70
:rem 11
830 DV=1-7*(A$="D"):IFDV=8THENF$="0:"+F$ :rem 142
:rem 142
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$) :rem 102
):POKE782,ZK/256 :rem 231
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN( :rem 206
T$):SYS65469 :rem 117
845 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 130
66 :rem 22
850 POKE780,0:SYS65493 :rem 119
860 IF(PEEK(783)AND1)OR(ST AND191)THEN870 :rem 119
:rem 111
865 PRINT"{GIU' }OPERAZIONE CONCLUSA.":GOT :rem 124
O310 :rem 119
870 PRINT"{GIU' }ERRORE DI CARICAMENTO.RIP :rem 102
ROVA.{GIU' }":IFDV=1THEN800 :rem 231
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINT#15 :rem 206
;E2$:CLOSE15:GOTO800 :rem 117
1000 REM CICALINO :rem 130
1001 POKE36878,15:POKE36874,190 :rem 22
1002 FORW=1TO300:NEXTW :rem 119
1003 POKE36878,0:POKE36874,0:RETURN :rem 119
:rem 74
2000 REM CAMPANELLO :rem 130
2001 FORW=1TO0STEP-1:POKE36878,W:POKE368 :rem 22
76,240:NEXTW :rem 119
2002 POKE36876,0:RETURN :rem 119
3000 PRINTC$;"{RVS}NON IN PAGINA ZERO O :rem 23
{ 2 SPAZI}SU ROM{ 4 GIU' }":GOTO1000 :rem 23

```

```

230 PRINT"[<5>]{ 2 SU} INDIRIZZO CONCLUSI :rem 183
VO{ 3 SPAZI}";:INPUTE:F=1-F:C$=CHR$( :rem 200
1+119*F) :rem 119
240 IFE<256OR(E>40960ANDE<49152)ORE>53247 :rem 119
THENGOSUB3000:GOTO230 :rem 179
250 IFE<STHENPRINTC$;"{RVS}INDIRIZZO CONC :rem 225
LUSIVO<INDIRIZZO INIZIALE" :rem 119
255 IFE<STHENGOSUB1000:GOTO230 :rem 179
260 PRINT:PRINT:PRINT :rem 225
300 PRINT"{CLR}";CHR$(14):AD=S:POKEV+21,0 :rem 227
:rem 33
310 A=1:PRINTRIGHT$( "0000"+MID$(STR$(AD), :rem 228
2),5);":FORJ=ATO6 :rem 62
315 FORJ=ATO6 :rem 64
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320 :rem 13
:rem 228
390 IFN=-211THEN710 :rem 62
400 IFN=-204THEN790 :rem 64
410 IFN=-206THENPRINT:INPUT"{GIU' } INSE :rem 13
RE IL NUOVO INDIRIZZO";ZZ :rem 105
414 IFN=-206THENIFZZ<SORZZ>ETHENPRINT" :rem 105
{RVS}ESCE DAL CAMPO DI VALORI INDICAT :rem 105
O" :rem 105

```

```

415 IFN=-206THENIFZZ<SORZZ>ETHENGOSUB1000
      GOTO410 :rem 202
417 IFN=-206THENAD=ZZ:PRINT:GOTO310
      :rem 238
420 IFN<>-196THEN480 :rem 133
430 PRINT:INPUT"LISTATO:DA";F:PRINT"
      { 9 SPAZI}A";:INPUTT :rem 183
440 IFF<SORE>EORT<SORT>ETHENPRINT"MINIMO"
      ;S;" MASSIMO";E;"!<5>":GOTO430
      :rem 147
450 FORI=FTOTSTEP6:PRINT:PRINTRIGHT$( "000
      0"+MID$(STR$(I),2),5);":": :rem 30
451 FORK=0TO5:N=PEEK(I+K):PRINTRIGHT$( "00
      "+MID$(STR$(N),2),3);":": :rem 66
460 GETA$:IFAS>" THENPRINT:PRINT:GOTO310
      :rem 25
470 NEXTK:PRINTCHR$(20);:NEXTI:PRINT:PRIN
      T:GOTO310 :rem 50
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK
      SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(146
      ); :rem 94
511 IFN=-1THENA=6:GOTO315 :rem 254
515 PRINTCHR$(20):IFN=CKSUMTHEN530
      :rem 122
520 PRINT:PRINT"{RED}LA LINEA E' STATA IN
      SERITA IN MANIERA" :rem 157
525 PRINT"ERRATA. RIPETERE[<5>]":PRINT:GO
      SUB1000:GOTO310 :rem 27
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT:POKE54
      272,0:POKE54273,0 :rem 227
550 AD=AD+6:IFAD<ETHEN310 :rem 212
560 GOTO710 :rem 108
570 N=0:Z=0 :rem 88
580 PRINT"[<~>": :rem 81
581 GETA$:IFAS=" THEN581 :rem 95
582 AV=- (AS="M")-2*(AS="")-3*(AS=".")-4*
      (AS="J")-5*(AS="K")-6*(AS="L")
      :rem 41
583 AV=AV-7*(AS="U")-8*(AS="I")-9*(AS="O"
      ):IFAS="H"THENA$="0" :rem 134
584 IFAV>0THENA$=CHR$(48+AV) :rem 134
585 PRINTCHR$(20);:A=ASC(AS):IFA=13ORA=44
      ORA=32THEN670 :rem 229
590 IFA>128THENN=-A:RETURN :rem 137
600 IFA<>20THEN630 :rem 10
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT"
      {SIN}{SIN}":GOTO690 :rem 172
620 GOTO570 :rem 109
630 IFA<48ORA>57THEN580 :rem 105
640 PRINTA$;:N=N*10+A-48 :rem 106
650 IFN>255THENA=20:GOSUB1000:GOTO600
      :rem 229
660 Z=Z+1:IFZ<3THEN580 :rem 71
670 IFZ=0THENGOSUB1000:GOTO570 :rem 114
680 PRINT",";:RETURN :rem 240
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211)
      :rem 149
691 FORI=1TO3:T=PEEK(S%-I) :rem 67
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT
      :rem 205
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN
      :rem 7
710 PRINT"{CLR}{RVS}*** SAVE ***
      { 3 GIU'}" :rem 236
715 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
      {OFF} PER USCIRE DAL SAVE{GIU'}"
      :rem 103

```

```

720 F$="":INPUT"{GIU'}NOME DEL FILE";F$:I
      FF$=" THENPRINT:PRINT:GOTO310
      :rem 42
730 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
      O {RVS}D{OFF}ISCO: (N/D)" :rem 128
740 GETA$:IFAS<>"N"ANDA$<>"D"THEN740
      :rem 30
750 DV=1-7*(AS="D"):IFDV=8THENF$="0":+F$:
      OPEN15,8,15,"S"+F$:CLOSE15 :rem 212
760 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
      ):POKE782,ZK/256 :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
      T$):SYS65469 :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654
      66 :rem 69
765 K=S:POKE254,K/256:POKE253,K-PEEK(254)
      *256:POKE780,253 :rem 17
766 K=E+1:POKE782,K/256:POKE781,K-PEEK(78
      2)*256:SYS65496 :rem 235
770 IF(PEEK(783)AND1)OR(191ANDST)THEN780
      :rem 111
775 PRINT"{GIU'}FATTO.{GIU'}":GOTO310
      :rem 201
780 PRINT"{GIU'}ERRORE NEL SAVE-RIPROVA!"
      :IFDV=1THEN720 :rem 104
781 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
      ;E2$:CLOSE15:GOTO720 :rem 103
790 PRINT"{CLR}{RVS}*** LOAD ***
      { 2 GIU'}" :rem 212
795 PRINT"{ 2 GIU'}PREMERE {RVS}RETURN
      {OFF} PER USCIRE DAL LOAD{GIU'}"
      :rem 96
800 F$="":INPUT"{ 2 GIU'}NOME DEL FILE";F
      $:IFF$=" THENPRINT:GOTO310 :rem 115
810 PRINT:PRINT"{ 2 GIU'}{RVS}N{OFF}ASTRO
      O {RVS}D{OFF}ISCO: (N/D)" :rem 127
820 GETA$:IFAS<>"N"ANDA$<>"D"THEN820
      :rem 28
830 DV=1-7*(AS="D"):IFDV=8THENF$="0":+F$:
      :rem 157
840 T$=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$
      ):POKE782,ZK/256 :rem 2
841 POKE781,ZK-PEEK(782)*256:POKE780,LEN(
      T$):SYS65469 :rem 107
845 POKE780,1:POKE781,DV:POKE782,1:SYS654
      66 :rem 70
850 POKE780,0:SYS65493 :rem 11
860 IF(PEEK(783)AND1)OR(191ANDST)THEN870
      :rem 111
865 PRINT"{GIU'}FATTO.":GOTO310 :rem 184
870 PRINT"{GIU'}ERRORE NEL LOAD-RIPETI!
      {GIU'}":IFDV=1THEN800 :rem 19
880 OPEN15,8,15:INPUT#15,E1$,E2$:PRINTE1$
      ;E2$:CLOSE15:GOTO800 :rem 102
1000 REM CICALINO :rem 231
1001 POKE54296,15:POKE54277,45:POKE54278,
      165 :rem 207
1002 POKE54276,33:POKE54273,6:POKE54272,5
      :rem 42
1003 FORT=1TO200:NEXT:POKE54276,32:POKE54
      273,0:POKE54272,0:RETURN :rem 202
2000 REM CAMPANELLO :rem 130
2001 POKE54296,15:POKE54277,0:POKE54278,2
      47 :rem 152
2002 POKE54276,17:POKE54273,40:POKE54272,
      0 :rem 86
2003 FORT=1TO100:NEXT:POKE54276,16:RETURN
      :rem 57
3000 PRINTCS$;"{RVS} NON IN PAGINA ZERO O
      SU{DES}ROM ":GOTO1000 :rem 240

```

7 Note Bit

Oggi la musica si impara così.



per C64

Il 1985 è l'anno mondiale della musica... preparati con **7 note Bit**, la nuova, fantastica opera che in soli 15 fascicoli quattordicinali (ciascuno corredato da una cassetta software per Commodore 64) ti insegna veramente la musica. E puoi suonare subito.

Non perdere il primo numero. Pensa, compresa nel prezzo c'è anche la stupenda tastiera professionale per il tuo Commodore 64.



**GRUPPO
EDITORIALE
JACKSON**

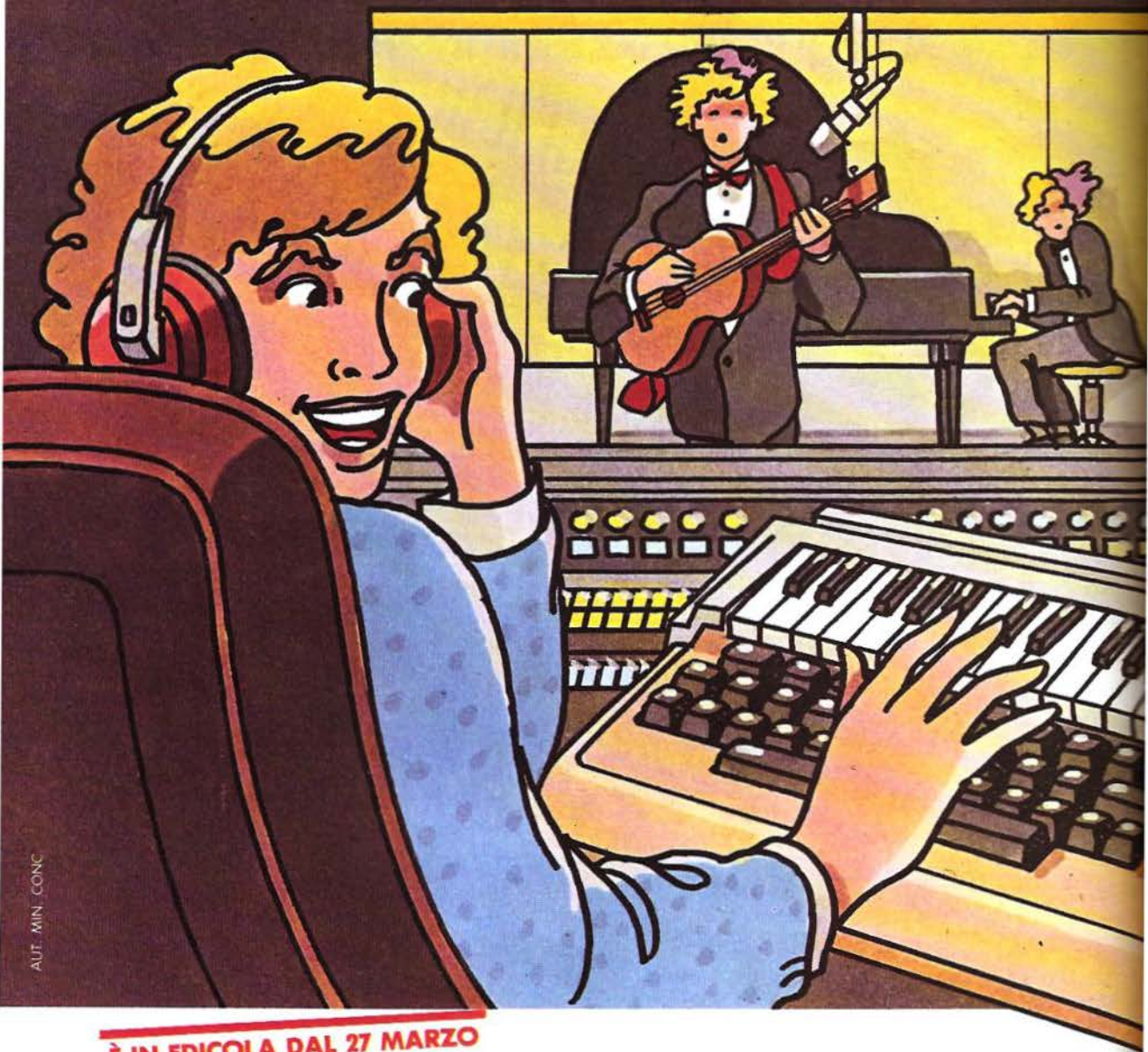
San Francisco-Londra-Milano

IN COLLABORAZIONE CON **SEL**

....E VINCI 30 STRAORDINARI
COMMODORE PLUS/4



7 NOTE BIT: OGGI LA M



AUT. MIN. CONC.

È IN EDICOLA DAL 27 MARZO
LA TASTIERA COMPRESA
NEL PRIMO FASCICOLO



Il 1985 è l'anno mondiale della musica... preparati con 7 note Bit, la nuova, fantastica opera che in soli 15 fascicoli quattordicinali (ciascuno corredato da una cassetta software per Commodore 64) ti insegna veramente la musica.

Con 7 note Bit, porti in casa il tuo maestro personale di musica che ti introduce alla teoria e al lessico musicali, ti insegna e ti fa fare pratica sulla tastiera, seguendoti passo passo nelle lezioni ... senza mai annoiarti.

E puoi suonare subito, con l'aiuto del software "interattivo" della cassetta e della speciale tastiera musicale per il tuo 64.

7 note Bit trasforma il Commodore 64 in uno strumento musicale sofisticato ed entusiasmante.

MUSICA SI IMPARA COSÌ.



Con 7 note Bit
vinci 30 straordinari
Commodore Plus/4



Il video del computer ti presenta tutto ciò che ti serve: un metronomo elettronico, il rigo musicale, la tastiera visualizzata con le note corrispondenti ai tasti ... e potrai partecipare "dal vivo" alle session dei musicisti più famosi, suonando insieme a loro.

Prenota in edicola **7 note Bit**: una prestigiosa iniziativa editoriale nata dall'unione della esperienza informatico-divulgativa del Gruppo Editoriale Jackson con la competenza tecnico-musicale **SIEL**.

7 note Bit è il "programma musicale" più interessante che c'è: non perdere il primo numero.

Pensa, compresa nel prezzo c'è anche la stupenda tastiera professionale per il tuo Commodore 64.



**GRUPPO
EDITORIALE
JACKSON**

REM:HW

Il Joystick

Come acquisire i segnali provenienti dai joystick e simularne il funzionamento sulla tastiera: parte terza

Come acquisire i segnali provenienti dal joystick del VIC 20

Paradossalmente, l'acquisizione dei segnali provenienti dall'unica Control Port del VIC 20, destinata esplicitamente al controllo dei joystick (un secondo joystick può essere interfacciato tramite la User Port), risulta più complicata delle analoghe operazioni sul C64. Innanzi tutto, anche se il VIC 20 destina un'unica interfaccia al controllo dei joystick, in realtà sono due — come per il C64, quindi — le locazioni riservate all'acquisizione dei segnali. Ma, mentre per il C64 le due locazioni controllano due unità distinte, per il VIC 20 esse servono un unico joystick. Come se ciò non bastasse, la suddivisione dei compiti tra le due locazioni coinvolte non è né equa né, apparentemente, razionale.

La prima delle due locazioni (37151) controlla infatti, oltre al pulsante di fuoco, gli spostamenti del joystick in senso verticale e, orizzontalmente, verso sinistra. La seconda (37152) controlla i soli spostamenti verso destra; ma non è tutto! Essa divide le sue funzioni tra il controllo degli spostamenti verso "EST", tramite il bit più significativo, e la scansione della tastiera. Ciò significa che non è possibile acquisirne il segnale senza disattivare, almeno provvisoriamente, parte della tastiera.

Ciò significa che le operazioni connesse con la lettura del segnale proveniente dal solo joystick connesso col VIC 20 sono molto più complesse e costose, sempre in termini di tempo, di quanto succede sul C64.

La logica del segnale è, come nel caso del fratello maggiore, "negativa"; la configurazione dei bit delle due locazioni è la seguente

Locazione 37151

Bit	7	6	5	4	3	2	1	0
Peso	128	64	32	16	8	4	2	1
	—	—	P	O	S	N	—	—

Locazione 37152

Bit	7	6	5	4	3	2	1	0
Peso	128	64	32	16	8	4	2	1
	E	—	—	—	—	—	—	—

Come detto, parte della locazione 37152 è riservata alla scansione della tastiera. Ciò significa che è possibile compiere una lettura semplice, ma non esaustiva, del segnale proveniente dal joystick, solo se non si desidera controllare gli spostamenti verso destra (ad esempio perché sono sufficienti i soli spostamenti in senso verticale).

di U.G. Barzaghi



L'acquisizione dei segnali connessi alla locazione 37151 è possibile tramite le istruzioni

```
***10 P=PEEK (37151) :P=NOTP
***20 IFPAND32THEN... : REM PUL-
    SANTE DI FUOCO
***30 IFPAND16THEN... : REM
    OVEST
***40 IFPAND8 THEN... : REM SUD
***50 IFPAND4 THEN... : REM
    NORD
```

Come potete vedere, il procedimento

non differisce da quanto fatto per l'acquisizione dell'analogo segnale del C64. L'unica differenza riguarda appunto gli spostamenti verso destra. Per avviare a questo inconveniente è necessario rinunciare al controllo di parte della tastiera. È ovviamente possibile farlo all'inizio del programma, tramite una istruzione

POKE 37154,127

rinunciando definitivamente al controllo di parte di essa, reinserendolo, magari prima della fine del programma, tramite una istruzione

POKE 37154,255

Questo nel caso in cui non sia indispensabile il controllo dell'intera tastiera nel corso del programma e l'acquisizione del segnale proveniente dal joystick (spostamenti verso destra inclusi) debba essere il più possibile rapida (ma in questo caso sarebbe probabilmente consigliabile far ricorso al linguaggio macchina).

Se è invece necessario avere il controllo dell'intera tastiera nel corso del programma, e non si hanno particolari problemi di tempo, potete utilizzare questa terna di istruzioni

***10 POKE 37154,127

***20 P=NOT (PEEK(37152)):IFPAND128THEN...:REM EST

***30 POKE 37154,255

Se siete curiosi di sapere quale sia la natura delle interferenze tra il joystick e la tastiera, provate ad inserire la presa del joystick nella Control Port del VIC e a premere i tasti numerici, tenendo il joystick spostato verso destra. Dato che i tasti della riga superiore sono tra i più utilizzati nel corso di un programma, capirete come sia consigliabile tenere ben distinte le due funzioni svolte dalla locazione 37152.

È ovviamente possibile concentrare la serie di istruzioni precedenti in qualche cosa di più pratico e di più facile gestione.

È possibile, ad esempio, far confluire la lettura delle due locazioni in una sola variabile, con una istruzione del tipo

***00 P= (NOT PEEK(37151)) AND 60 - ((PEEK(37152) AND 128) = 0)

decodificandola quindi nel modo consueto

***10 IFPAND32:THEN...:REM PULSANTE DI FUOCO

***20 IFPAND16:THEN...:REM OVEST

***30 IFPAND 8:THEN...:REM SUD

***40 IFPAND 4:THEN...:REM NORD

***50 IFPAND 1:THEN...:REM EST

Come avrete potuto notare, la seconda parte della istruzione in ***00 traduce

la lettura del segnale in logica "negativa", proveniente dal bit più significativo della locazione 37152, in un segnale in logica "positiva", che coinvolge il bit meno significativo della locazione rappresentata dalla variabile P.

Un altro modo di acquisire lo stesso segnale letto in ***00 consiste nello scomporlo nelle sue componenti orizzontali e verticali tramite la seguente coppia di istruzioni

OZ = SGN(X AND 1) - SGN(X AND 16)

VT = SGN(X AND 8) - SGN(X AND 4)

Il contenuto della variabile OZ (non è possibile utilizzare una variabile avente per iniziale la coppia di lettere OR per indicare gli spostamenti orizzontali, dal momento che l'interprete BASIC non le accetterebbe come nome di variabile - anche se seguite da altre lettere - data l'esistenza dell'operatore logico OR) sarà uguale a +1, -1 o 0 in base al fatto che siano attivati, rispettivamente, i microinterruttori di destra, sinistra o nessuno dei due. Analogamente, la variabile VT conterrà un valore numerico pari a +1 nel caso in cui il joystick venga spostato verso l'alto e -1 nel caso in cui lo spostamento sia stato verso il basso.

Esiste, però, un modo ancora più elegante di acquisire e convertire i segnali provenienti dalle due locazioni connesse con la Control Port, pulsante di fuoco escluso. Esso può essere raccolto nel breve programma e nella tabella che seguono

***00 POKE 37154,127

***10 V1=PEEK(37152) AND 128

***20 V2=PEEK(37151) AND 28

***30 POKE 37154,255

***40 JS=V1/16+V2/4

***50 JS=(NOTJS)AND15

Posizione del joystick	JS	DX	DY
Neutra	0	0	0
Nord	1	0	-1
Sud	2	0	1
Impossibile	3	0	0
Ovest	4	-1	0
Nord-Ovest	5	-1	-1
Sud-Ovest	6	-1	1
Impossibile	7	0	0
Est	8	1	0
Nord-Est	9	1	-1
Sud-Est	10	1	1

L'istruzione di riga ***40 del breve programma precedente congloba i segnali provenienti dalle due locazioni in un unico byte (quello riservato alla variabile JS) con la seguente configurazione dei bit riservati al controllo dei singoli microinterruttori

Bit	7	6	5	4	3	2	1	0
Peso	128	64	32	16	8	4	2	1
	-	-	-	-	E	O	S	N

Dividendo per 16 il segnale (contenuto nella variabile V1) proveniente dal bit più significativo della locazione 37152, lo si trasferisce, infatti, dal bit di peso 128 al bit di peso 128/16=8.

Sommando questo segnale a quello proveniente dalla locazione 37151 (contenuto nella variabile V2), diviso a sua volta per 4 (e traslato perciò di due bit verso destra), si ottiene la configurazione suindicata in cui i quattro bit di controllo del segnale offrono il vantaggio di consentire una configurazione più simile a quella vista per le locazioni connesse con la Control Port del C64.

L'unico inconveniente è presentato dall'assenza del pulsante di fuoco. Se si è disposti a rinunciare alla rapida decodifica del segnale di spostamento, riassunta dalla tabella precedente (dove vengono anche riportate le variazioni delle coordinate di un eventuale oggetto controllato tramite il joystick), è possibile inserirlo in maniera piuttosto semplice, sostituendo alle istruzioni ***40 e ***50 le seguenti

***40 JS=V1/8+V2/4

***50 JS=(NOTJS)AND 31

ottenendo la configurazione

Bit	7	6	5	4	3	2	1	0
Peso	128	64	32	16	8	4	2	1
	-	-	-	E	P	O	S	N

dove P rappresenta il bit connesso col microinterruttore del pulsante di fuoco.

Oppure, se si vogliono mantenere i segnali di spostamento in un campo di valori utilizzabili come indici di eventuali matrici, è necessario aggiungere l'istruzione

***15 V3=PEEK(37151)AND32

e modificare le istruzioni ***40 e ***50 nel modo seguente

***40 JS=V1/16+V2/4+V3/2

***50 JS=(NOTJS)AND31

in modo da ottenere la configurazione

Bit	7	6	5	4	3	2	1	0
Peso	128	64	32	16	8	4	2	1
	-	-	-	P	E	O	S	N

che consente di utilizzare la tabella precedente.

Programma 1: funzionamento e commenti al listato

Dato il carattere prettamente didattico di questo articolo e dei programmi che lo corredano, abbiamo preferito rinunciare ad utilizzare, per quanto riguarda il VIC 20, eventuali espansioni di memoria, per dar modo anche agli utenti

REM:HW

di VIC 20 in configurazione standard (inespansa) di utilizzare i programmi studiati per il loro calcolatore.

Ciò ha, ovviamente, richiesto un paziente lavoro di "distillazione" delle istruzioni e la rinuncia a parecchie delle opzioni offerte dai programmi per il C64.

La prima, e più vistosa, differenza è data dalla indisponibilità sul VIC 20 degli sprite. Per supplire a questa mancanza abbiamo pensato di utilizzare i caratteri grafici personalizzati. Questa scelta ci ha posto di fronte ad un'altra serie di problemi, innanzi tutto in termini di occupazione di memoria. La memoria RAM (Random Access Memory, memoria ad accesso casuale) del calcolatore, infatti, doveva contenere non solo i dati necessari a ridefinire gli otto caratteri destinati a contenere le frecce della "bussola", ma anche lo spazio riservato ad una copia parziale del set di caratteri contenuto in memoria ROM (Read Only Memory, memoria a sola lettura). Un ulteriore problema era rappresentato dalla scelta di quali caratteri dovessero far parte della copia. Il loro numero doveva, ovviamente, essere limitato (la definizione di ogni carattere richiede 8 byte), ma il set doveva, necessariamente, coprire almeno la serie delle lettere maiuscole dell'alfabeto, le dieci cifre ed una serie di simboli aritmetici (=, +, -, /). Era inoltre necessario prevedere dello spazio disponibile a contenere i caratteri ridefiniti e disporre questi ultimi in modo che il loro controllo fosse pratico, il più possibile semplice, ma senza rinunciare ad alcuno dei caratteri suddetti né occupare una quantità di memoria eccessiva.

Oltre a ciò, lo spazio riservato al set di caratteri personalizzato andava protetto da eventuali interferenze del listato BASIC, abbassando il limite superiore della memoria RAM riservata a contenerlo.

Tutto ciò significa che, per quanto riguarda il programma 1, l'uscita su video del quadro della bussola risulta, ad esempio, molto semplificata rispetto a quanto visto per il C64.

Dopo aver copiato il programma, ed averne registrato una copia, è possibile eseguirlo tramite un RUN. Dato che la Control Port cui collegare il joystick è, in configurazione standard, una sola, non è stato necessario inserire alcun menu di scelta prima dell'uscita del piano della bussola.

Sarà comunque necessario attendere che il calcolatore copii dalla ROM parte del set di caratteri e vi inserisca la ridefinizione di quelli tra essi che verranno utilizzati per indicare le varie direzioni sulla bussola.

Un'ulteriore, gravosa, limitazione è da-

ta dalle dimensioni dello schermo del VIC 20. Per questa ragione il quadro delle direzioni è volutamente abbreviato (solo le quattro direzioni cardinali vengono indicate per esteso, le direzioni intermedie sono identificate dalle sole iniziali) ed anche le indicazioni sulla parte superiore dello schermo sono assai ridotte. Nell'angolo superiore sinistro, in blu, viene visualizzato il valore della variabile P, pari al valore numerico acquisito tramite la Control Port, mentre nell'angolo superiore destro, in verde, viene indicato il valore della variabile JS, ricavata dall'istruzione

$JS = ABS((P-100)/4)-7$

Spostando in qualsiasi direzione la leva del joystick comparirà una freccia puntata nella direzione corrispondente e la direzione indicata verrà evidenziata in rosso, mentre il contenuto delle variabili P e JS verrà aggiornato. Premendo il pulsante del joystick è possibile cambiare il colore del carattere al centro dello schermo. Per interrompere il programma è necessario premere RUN/STOP.



Come simulare un joystick tramite la tastiera del VIC

Le differenze esistenti fra l'acquisizione dei segnali provenienti dal joystick del C64 rispetto al VIC 20 sono, come abbiamo avuto modo di vedere nei due paragrafi precedenti, decisamente notevoli.

Volendo utilizzare, invece, la tastiera del VIC 20 per simulare il funzionamento di un joystick, non si rilevano differenze di particolare entità rispetto a

Il Joystick

quanto visto per il C64, tanto che non risulta particolarmente difficile convertire per uso sul VIC 20 un programma che funzioni su C64 e che utilizzi come controllo non il joystick, ma la tastiera (a parte, ovviamente, tutte le diversità esistenti tra i due calcolatori per dimensioni e posizionamento della memoria di schermo o della memoria di controllo del colore). Pertanto riteniamo superfluo ripetere quanto già diffusamente trattato nell'articolo di commento al programma per C64; ci limiteremo invece ad evidenziare le diversità esistenti.

Per quanto riguarda l'acquisizione dei segnali tramite istruzioni GET non esistono differenze tra i due calcolatori, dal momento che i rispettivi codici ASCII sono identici. Anche la locazione da utilizzare per acquisire segnali tramite istruzioni PEEK è la stessa (197); cambiano però i valori numerici che si ottengono in corrispondenza degli stessi tasti.

Come nel Programma per C64, pubblicato sul numero scorso della rivista, anche nel Programma 2 abbiamo comunque inserito una breve routine, che consente di visualizzare sullo schermo i valori numerici corrispondenti ai tasti premuti ed acquisiti mediante l'istruzione PEEK (197). È possibile attivare questo breve sottoprogramma battendo

Run 950

Come ricorderete, premendo RETURN il valore ottenuto sul C64 era pari a 60. Lo stesso tasto, premuto sul VIC 20 ed acquisito con la stessa istruzione PEEK (197), restituisce un valore pari a 32. Analoghe differenze sono presenti su quasi tutti gli altri tasti. La tabella seguente elenca i valori numerici ottenuti tramite l'istruzione PEEK (197) sul C64 e sul VIC 20 per i tasti, utilizzati nei due programmi, dedicati alla simulazione del funzionamento di un joystick sulla tastiera.

Valore ottenuto

tramite l'istruzione PEEK (197)

	C64	VIC 20
Tasto RETURN	1	15
Barra spaziatrice	60	32
f1	4	39
f7	3	63
Q	62	48
W	9	9
E	14	49
A	10	17
D	18	18
Z	12	33
X	23	26
C	20	34

Non dovrebbero esserci particolari pro-

OK-List per la perfetta battitura dei listati

Inserite in memoria l'OK-LIST prima di accingervi a battere i listati pubblicati: avrete a disposizione un formidabile mezzo per essere sicuri di aver inserito nel computer ogni carattere in modo assolutamente corretto.

Tutti i nostri listati sono caratterizzati da un numero aggiunto ad ogni linea di programma, ad esempio: rem 123. **NON COPIATE QUESTA PARTE DELLA LINEA!** Serve unicamente per vostra informazione. L'istruzione REM ha l'unico scopo di renderla inoffensiva, se per errore dovete batterla.

Se, prima di iniziare il lavoro di battitura del listato, caricate in memoria OK-LIST e lo attivate con RUN, per ogni linea che battete verrà visualizzato un numero (detto checksum) nell'angolo superiore sinistro dello schermo.

Confrontate questo numero con quello pubblicato per ogni linea del listato: se non corrisponde, ciò significa che avete commesso un errore nella battitura della linea stessa.

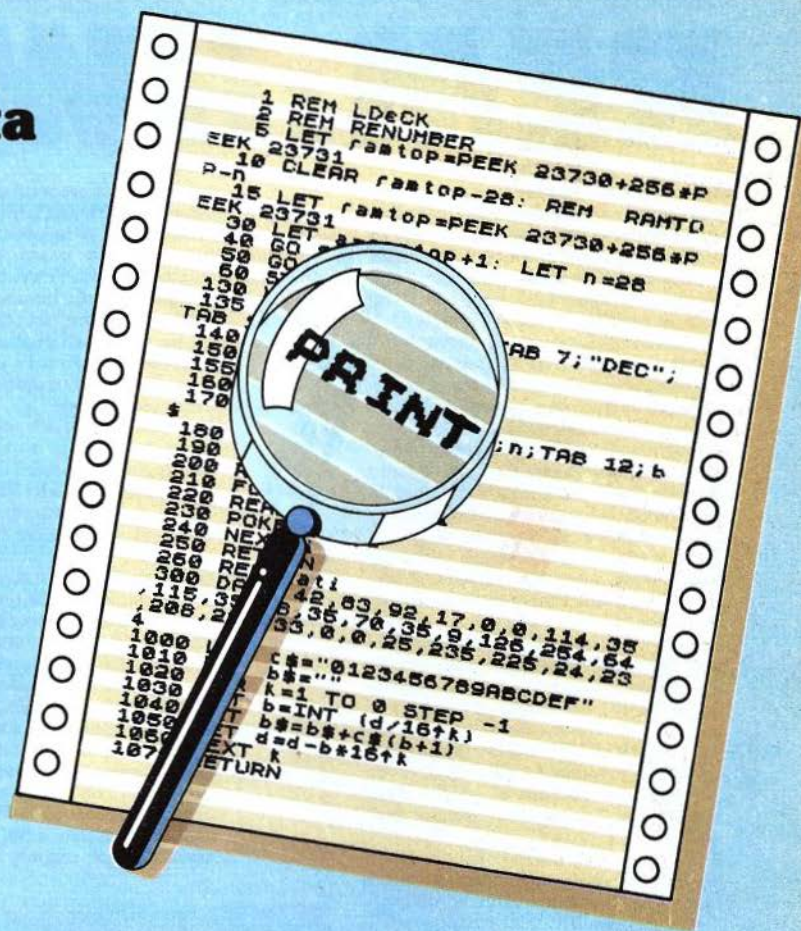
Le ore buttate per cercare e correggere errori di battitura in programmi che non funzionano saranno solo un lontano ricordo!

Due ultime cose di cui tenere conto: OK-LIST non tiene conto degli spazi: questo per vostra convenienza, poiché questi generalmente sono poco importanti.

Infine, OK-LIST è allocato nel buffer del registratore (locazioni 886-1018), quindi, prima di salvare su cassetta il programma che state battendo, ricordatevi di disabitarlo premendo contemporaneamente RUN/STOP + RESTORE. Potrete poi riattivarlo con SYS 886.

Il sistema più comodo per usare OK-LIST è quello di battere normalmente il programma, quindi listarlo e portarsi con il cursore sul numero di linea più basso che appare sul video; premere quindi RETURN e confrontare il numero di check-sum visualizzato nell'angolo superiore sinistro del video con quello pubblicato per la data riga del programma. Il cursore si porterà automaticamente sulla linea successiva e non dovrete fare altro che premere il checksum seguente.

Data l'area di memoria in cui si trova, OK-LIST non può essere usato per controllare un programma già salvato su cassetta, mentre ciò è possibile se il programma è stato registrato su disco.



OK-LIST versione per VIC 20 e C 64

```

100 PRINT"{CLR}ATTENDERE PREGO..."
105 FORI=886TO1018:READA:CK=CK+A:POKEI,A:
NEXT
110 IF CK<>17539 THEN PRINT"{GIU'}ERRORE
NELLE ISTRUZIONI DATA":END
120 SYS886:PRINT"{CLR}{" 2 GIU'}OK-LIST AT
TIVATO.":NEW
886 DATA 173,036,003,201,150,208
892 DATA 001,096,141,151,003,173
898 DATA 037,003,141,152,003,169
904 DATA 150,141,036,003,169,003
910 DATA 141,037,003,169,000,133
916 DATA 254,096,032,087,241,133
922 DATA 251,134,252,132,253,008
928 DATA 201,013,240,017,201,032
934 DATA 240,005,024,101,254,133
940 DATA 254,165,251,166,252,164
946 DATA 253,040,096,169,013,032
952 DATA 210,255,165,214,141,251
958 DATA 003,206,251,003,169,000
964 DATA 133,216,169,019,032,210
970 DATA 255,169,018,032,210,255
976 DATA 169,058,032,210,255,166
982 DATA 254,169,000,133,254,172
988 DATA 151,003,192,087,208,006
994 DATA 032,205,189,076,235,003
1000 DATA 032,205,221,169,032,032
1006 DATA 210,255,032,210,255,173
1012 DATA 251,003,133,214,076,173
1018 DATA 003

```

REM:HW

Il joystick

blemi anche con le altre locazioni da noi indicate come fonti di segnali connessi con la tastiera, anche se confessiamo di non averle provate per quanto riguarda il VIC 20; riteniamo, comunque, che per quanto riguarda il controllo dello stato di SHIFT, CTRL o COMMODORE non dovrebbero esserci rilevanti differenze, mentre immaginiamo che la locazione 203 presenti le stesse differenze esistenti tra i due segnali della 197 sui due diversi calcolatori.

Il Programma 2 funziona in modo analogo al Programma per C64 per quanto riguarda l'uscita del piano della bussola e la freccia indicatrice visualizzata.

Contrariamente al programma precedente in versione VIC, però, abbiamo mantenuto il menu che consente di scegliere, come sul Programma 2, tra acquisizione del segnale tramite PEEK(197) e tramite GET X\$. Anche i tasti di controllo sono gli stessi: il tasto fl consente il passaggio da una opzione del menu alla precedente ed il tasto f7 realizza il trasferimento in senso inverso. Il tasto RETURN e la barra spaziatrice, infine, fungono da pulsanti di fuoco "virtuali" per passare al quadro della bussola una volta scelta l'opzione desiderata.

Il fatto di fornire questo menu di scelta ha fatto sì, però, che avessimo ancor meno spazio disponibile nella memoria inespressa del VIC 20, costringendoci ad un minuzioso lavoro per quanto riguarda, ad esempio, la scelta dei caratteri da ridefinire.

Il trasferimento del set di caratteri ROM in RAM, operato dal sottoprogramma delle righe 600-660, si limita ai primi 61 caratteri; dovendo inoltre disporre, per la visualizzazione del menu, di alcuni dei caratteri che nel programma precedente venivano ridefiniti ('\$', ' ', ' ', ecc.), è stato necessario personalizzare altri caratteri e, per ragioni di spazio, gli otto caratteri contenenti le frecce indicatrici sono così risultati non consecutivi. Questo fatto ci ha costretti a memorizzare l'indirizzo dei vari indicatori sotto forma di scarto da sommare ad un indirizzo di base comune.

Nelle righe seguenti vengono commentate, brevemente, le istruzioni fondamentali che differenziano il Programma per C64 dal Programma 1.

Righe
30-100

Commento

Visualizzazione del menu e inizializzazione delle variabili ad esso relative, in maniera non diversa da quanto fatto sul C64.

110

Acquisizione del tasto premuto tramite istruzione PEEK(197). Il test stabi-

120-130

230

250

isce se è stato premuto uno dei due pulsanti di fuoco "virtuali"; in caso positivo il programma passa alla sezione di visualizzazione del quadro della bussola.

Si compiono test per stabilire se siano stati premuti i tasti funzione fl o f7 e si aggiorna di conseguenza l'opzione attivata.

Nel caso in cui sia stata scelta la seconda opzione per quanto riguarda l'acquisizione dei dati (GET X\$) si dirama verso la sezione di programma opportuna.

Si acquisisce il segnale proveniente dalla tastiera tramite l'istruzione PEEK(197) (prima opzione) e si azzerla la variabile FL, che indica la significatività del segnale acquisito. Si richiama, quindi, il sottoprogramma di decodifica del segnale. Si provvede ad eseguire un test sul contenuto della variabile FL e nel caso in cui esso sia uguale a 1 - segnale non

290

320-335

320

360-370

360

370

valore di base del codice di carattere (27) e viene quindi inserito nella locazione corrispondente della memoria di schermo. La differenza rispetto a quanto fatto in riga 260 del programma 1 è data dal fatto che, in questo caso, la non consecutività dei caratteri personalizzati non consente di utilizzare come scarto il valore SP (utilizzato anche come indice delle matrici contenenti i dati della bussola), ma rende necessaria l'introduzione di una seconda variabile JS.

Viene visualizzato il contenuto attuale della variabile P e X\$.

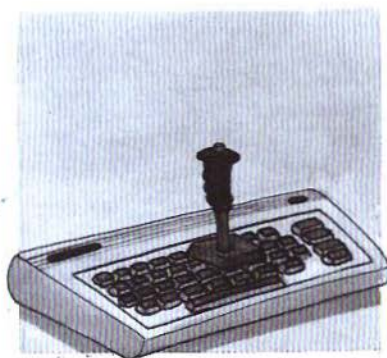
Sottoprogramma di controllo del colore del carattere visualizzato al centro dello schermo (segnale acquisito tramite PEEK(197)). In particolare:

Il test tende a stabilire se sia stato premuto uno dei due pulsanti di fuoco virtuali definiti sulla tastiera. In caso negativo il sottoprogramma si interrompe.

Controllo del colore della freccia indicatrice con acquisizione del segnale tramite istruzione GET X\$. In particolare:

Viene acquisito il carattere corrispondente al tasto premuto. Se non è stato premuto alcun tasto, si richiama.

Il carattere alfanumerico acquisito viene convertito nel corrispondente codice ASCII. Il test successivo stabilisce se sia stato premuto uno dei due pulsanti di fuoco virtuali; se il test ha esito positivo, richiama la parte del sottoprogramma di controllo del colore comune alle due opzioni. In caso negativo il segnale acquisito viene utilizzato per il controllo della direzione del joystick "virtuale", in maniera non dissimile da quanto fatto nel programma 2, per il C64. La subroutine di decodifica del segnale viene richiamata in riga 380 e, se il segna-



280

significativo - si provvede a richiamare il sottoprogramma di controllo del colore del carattere.

Nel caso in cui il segnale acquisito dal joystick sia significativo, si inserisce, tramite una istruzione POKE, il colore attuale nella locazione della memoria colore corrispondente al centro dello schermo. Il carattere da visualizzare viene stabilito aggiungendo uno scarto opportuno (JS) ad un

le è significativo, in riga 280 ci si ricollega alla parte del programma che si occupa di aggiornare la visualizzazione del piano della bussola sullo schermo. N.B. - Questa caratteristica, non molto elegante, configurazione "a spaghetti", ricca cioè di salti condizionati e attivazione di sottoprogrammi, è resa indispensabile dalle limitate disponibilità di spazio in memoria.

Vengono trasferiti in memoria RAM (a partire dalla locazione 7168) i dati che

definiscono i primi 496/8=62 caratteri del set contenuto in ROM.

Dati di ridefinizione dei caratteri "personalizzati". In questo caso i caratteri ridefiniti non sono consecutivi.

Sottoprogramma di decodifica del segnale acquisito. Per ragioni, ancora una volta, di economia di spazio è stato indispensabile concentrare la decodifica in un unico sottoprogramma per entrambe le opzioni, operando test su entrambi i set di valori acqui-

siti, rallentando così la velocità del programma.

Come per il Programma 2 questa riga non viene mai richiamata nel corso del programma e potrebbe, pertanto, essere tralasciata. Può essere eseguita tramite un comando in formato diretto RUN950 e risultare, però, utile nel caso in cui si desideri conoscere i valori numerici corrispondenti all'acquisizione dei tasti tramite l'istruzione PEEK(197); valori che non vengono elencati nel "Manuale d'uso del VIC 20".

600

programma 1 per VIC 20

```

5 POKE56,28:POKE52,28:CLR:CO=2 :rem 33
10 RS="{HOME}{ 22 GIU' }":DIMWS(22),SP(7,2) :rem 52
   ,SP$(7):FORI=1TO22
20 WS(I)=LEFT$(RS,I):NEXT:FORI=0TO7:FORJ=
   0TO2:READSP(I,J):NEXT:READSP$(I):NEXT :rem 200
30 POKE36879,25:PRINT"{CLR}":GOSUB6000 :rem 84
   :rem 18
150 PS=RIGHT$( "{ 3 SPAZI }"+STR$(M(NS,1)),3) :rem 16
170 PRINT"{CLR}"TAB(2)"{BLU}P="TAB(15)" :rem 168
   {GRN}JS=" :rem 16
210 FORI=0TO7:PRINTWS(SP(I,1))TAB(SP(I,2)) :rem 168
   )" {BLK}"SP$(I):NEXT :rem 168
250 POKE37154,127:P=(PEEK(37137)AND28)OR(
   PEEK(37152)AND128):POKE37154,255 :rem 109
255 FL=0:GOSUB770:IFFL=1THENGOSUB320:GOTO :rem 10
   250 :rem 10
260 POKE7932,35+SP:POKE38652,CO:POKE7932, :rem 46
   35+SP:JS=ABS((P-100)/4)-7 :rem 46
280 PS=RIGHT$(STR$(P),3):JS$=RIGHT$( :rem 158
   { 2 SPAZI }"+STR$(JS),2) :rem 158
290 PRINTWS(1)TAB(4)"{BLU}"P$TAB(18)" :rem 12
   {GRN}"JS$ :rem 12
300 IFOS<>SPTHENPRINTWS(SP(OS,1))TAB(SP(OS, :rem 6
   2))" {BLK}"SP$(OS) :rem 6
310 PRINTWS(SP(SP,1))TAB(SP(SP,2))" {RED}" :rem 223
   SP$(SP):OS=SP:GOTO250 :rem 223
320 B=PEEK(37137)AND32:IFB<>0THENRETURN :rem 139
   :rem 82
340 CO=CO+1:IFCO=8THENCO=0 :rem 91
345 IFCO=1THENCO=2 :rem 91
350 POKE38652,CO:FORI=1TO500:NEXT:RETURN :rem 48
410 DATA6,4,9,"NORD",5,20,9,"SUD",3,12,2, :rem 228
   "OVEST",11,12,14,"EST",1,17,4,"S-O" :rem 122
420 DATA2,8,4,"N-O",12,8,13,"N-E",20,17,1 :rem 171
   3,"S-E" :rem 178
770 IFP=152THENSP=0:RETURN :rem 172
780 IFP=148THENSP=1:RETURN :rem 122
790 IFP=140THENSP=2:RETURN :rem 168
800 IFP=28THENSP=3:RETURN :rem 174
810 IFP=132THENSP=4:RETURN :rem 124
820 IFP=136THENSP=5:RETURN :rem 122
830 IFP=24THENSP=6:RETURN :rem 183
840 IFP=20THENSP=7:RETURN :rem 234
850 FL=1:RETURN :rem 24
6000 FORI=0TO512:POKE7168+I,PEEK(32768+I) :rem 24
   :NEXT :rem 24
6010 READX:IFX<0THENPOKE36869,255:RETURN :rem 24

```

950

```

6020 FORI=XTOX+7:READJ:POKEI,J:NEXT:GOTO6 :rem 101
   010 :rem 101
6030 DATA7448,16,56,124,16,16,16,56,40,74 :rem 86
   56,40,56,16,16,16,124,56,16:rem 86
6040 DATA7464,0,0,32,99,254,99,32,0,7472, :rem 40
   0,0,4,198,127,198,4,0 :rem 40
6050 DATA7480,0,4,14,12,144,224,224,240,7 :rem 149
   488,240,224,224,144,12,14,4,0 :rem 149
6060 DATA7496,15,7,7,9,48,112,32,0,7504,0 :rem 74
   ,32,112,48,9,7,7,15,-1 :rem 74

```

programma 2 per VIC 20

```

5 POKE52,28:POKE56,28:CLR :rem 230
10 RS="{HOME}{ 22 GIU' }":DIMWS(22),SP(7,2) :rem 52
   ,SP$(7):FORI=1TO22
20 WS(I)=LEFT$(RS,I):NEXT:FORI=0TO7:FORJ= :rem 200
   0TO2:READSP(I,J):NEXT:READSP$(I):NEXT :rem 200
30 CO=2:POKE36879,25:PRINT"{CLR}":GOSUB60 :rem 80
   0:PRINT"{CLR}"TAB(7)"{RED}M E N U'":NS :rem 39
   =2:M$(1)="PEEK(197)" :rem 39
80 M(1,1)=41:M(2,1)=83:M$(2)="GET X$" :rem 80
   :rem 80
90 FORI=1TO2:PRINTWS(5+I*2)TAB(3)"{RED} :rem 95
   {RVS}"RIGHT$(STR$(I),1)"{OFF}{BLK} - " :rem 95
   M$(I):NEXT :rem 95
100 PRINTWS(5+NS*2)TAB(7)"{RED}"M$(NS):PR :rem 255
   INTWS(5+VS*2)TAB(7)"{BLK}"M$(VS) :rem 255
110 P1=PEEK(197):IFP1=15ORP1=32THEN170 :rem 223
   :rem 223
120 IFP1=39THENVS=NS:NS=NS-1:IFNS=0THENNS :rem 47
   =2 :rem 47
130 IFP1=63THENVS=NS:NS=NS+1:IFNS=3THENNS :rem 45
   =1 :rem 45
140 GOTO100 :rem 95
170 PS=RIGHT$( "{ 3 SPAZI }"+STR$(M(NS,1)), :rem 227
   2):PRINT"{CLR}"TAB(2)"{BLU}P="PS;TAB( :rem 212
   15)" {GRN}X$={BLK}" :rem 212
210 FORI=0TO7:PRINTWS(SP(I,1))TAB(SP(I,2)) :rem 252
   )SP$(I):NEXT :rem 252
230 IFNS=2THEN360 :rem 227
250 P=PEEK(197):FL=0:GOSUB770:IFFL=1THENG :rem 227
   OSUB320:GOTO250 :rem 227
280 POKE38652,CO:POKE7932,27+JS:P$=RIGHT$( :rem 147
   "{ 3 SPAZI }"+STR$(P),2) :rem 147
290 PRINTWS(1)TAB(4)"{BLU}"P$;TAB(18)" :rem 2
   {GRN}"X$ :rem 2
300 IFOS<>SPTHENPRINTWS(SP(OS,1))TAB(SP(OS, :rem 6
   2))" {BLK}"SP$(OS) :rem 6
310 PRINTWS(SP(SP,1))TAB(SP(SP,2))" {RED}" :rem 221
   SP$(SP):OS=SP:GOTO230 :rem 221

```

Per ricevere le riviste arretrate
di **SUPERCOMMODORE** con cassetta
compilate e spedite il coupon qui sotto!

SUPERCOMMODORE

CEDOLA DI ORDINAZIONE RIVISTE ARRETRATE CON CASSETTA ALLEGATA
da compilare e spedire in busta chiusa a
J. soft - Viale Fratelli, 5 - 20124 Milano - Tel. 6989228

BUONO D'ORDINE PER RIVISTE ARRETRATE
Anche se l'ordine riguardasse la sola cassetta questa
verrà comunque inviata insieme alla rivista al prezzo in-
dicato.

Vogliate inviarmi i numeri
di **SUPERCOMMODORE** con cassetta Anno

Al prezzo di L. 15.000

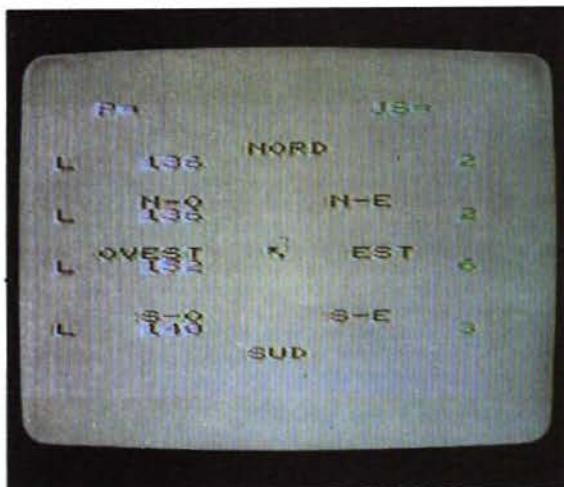
☐ Contanti allegati ☐ Assegno allegato n°

☐ No spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul cc. n° 19445204 intestato a J. soft - Milano

☐ Pagherò in contassegno al postino al ricevimento

Nome
Cognome
Via
Città C.A.P. Prov.
Se richiedo fatture:
Cod. F. e P. Iva
Data
Firma
Per i residenti all'estero - pagamento anticipato (vaglia o versamento su r/c cc)



```

320 IFP<>15ANDP<>32THENRETURN      :rem 101
340 CO=CO+1:IFCO=8THENCO=0           :rem 82
345 IFCO=1THENCO=2                   :rem 91
350 POKE38652,CO:FORI=1TO500:NEXT:RETURN      :rem 48
360 GETX$:IFX$=""THEN360              :rem 131
370 P=ASC(X$):IFP=13ORP=32THENGOSUB340:GO   :rem 106
   TO360                               :rem 106
380 FL=0:GOSUB770:IFFL=1THENGOTO360        :rem 188
                                           :rem 111
390 GOTO280                             :rem 111
410 DATA6,4,9,"NORD",5,20,9,"SUD",3,12,2,  :rem 228
   "OVEST",11,12,14,"EST",1,17,4,"S-O"      :rem 228
420 DATA2,8,4,"N-O",12,8,13,"N-E",20,17,1  :rem 122
   3,"S-E"                                   :rem 122
600 FORI=0TO496:POKE7168,I,PEEK(32768+I):   :rem 197
   NEXT                                     :rem 197
610 READX:IFX<0THENPOKE36869,255:RETURN     :rem 232
                                           :rem 232
620 FORI=XTOX+7:READJ:POKEI,J:NEXT:GOTO61  :rem 5
   0                                         :rem 5
630 DATA7384,16,56,124,16,16,16,56,40,739  :rem 36
   2,40,56,16,16,16,124,56,16              :rem 36
640 DATA7400,0,0,32,99,254,99,32,0,7408,0   :rem 237
   ,0,4,198,127,198,4,0                     :rem 237
650 DATA7416,0,4,14,12,144,224,224,240,74  :rem 89
   32,240,224,224,144,12,14,4,0             :rem 89
660 DATA7448,15,7,7,9,48,112,32,0,7464,0   :rem 28
   32,112,48,9,7,7,15,-1                    :rem 28
770 IFP=9ORP=87THENSP=0:X$="W":JS=0:RETUR  :rem 187
   N                                         :rem 187
780 IFP=26ORP=86THENSP=1:X$="X":JS=1:RETU  :rem 239
   RN                                        :rem 239
790 IFP=17ORP=65THENSP=2:X$="A":JS=2:RETU  :rem 214
   RN                                        :rem 214
800 IFP=18ORP=66THENSP=3:X$="D":JS=3:RETU  :rem 215
   RN                                        :rem 215
810 IFP=33ORP=90THENSP=4:X$="Z":JS=4:RETU  :rem 232
   RN                                        :rem 232
820 IFP=48ORP=81THENSP=5:X$="Q":JS=5:RETU  :rem 233
   RN                                        :rem 233
830 IFP=49ORP=69THENSP=6:X$="E":JS=6:RETU  :rem 232
   RN                                        :rem 232
840 IFP=34ORP=67THENSP=7:X$="C":JS=7:RET  :rem 9
   URN                                      :rem 9
850 FL=1:RETURN                             :rem 183
950 PRINT"{CLR}":FORI=1TO1000:PRINT"      :rem 27
   {HOME}"RIGHT$(("{ 4 SPAZI}"+STR$(PEEK(
   197)),3):NEXT

```

Othello per C16

di A. Motta

Questo programma è la rielaborazione della prima versione fatta per il personal ZX81 ed apparsa sulla rivista Personal Software n. 17 e n. 18.

Rispetto a tale versione, scritta in gran parte in linguaggio macchina a seguito della lentezza del BASIC del succitato personal, il programma per il C 16 è fatto interamente in BASIC e contiene diversi miglioramenti sia dal punto di vista grafico che sonoro e di gestione del gioco. Rimane invariato lo schema di gioco del computer ad entrambi i livelli di abilità, mentre, a parità di valore fra più mosse possibili, sceglie a caso quella da effettuare, mentre nella prima versione sceglieva sempre l'ultima. Inoltre, l'inserimento della mossa avviene mediante coordinate (prima si spostava una pedina che lampeggiava sullo schermo) ed il computer è in grado di stabilire se il giocatore può effettuare la mossa successiva o deve passare il turno. La possibilità di avere a disposizione i colori e il suono rende il tutto più interessante. Per i primi abbiamo scelto quelli classici del gioco stesso: nero e bianco per le pedine, verde per la scacchiera. Per quando riguarda il suono le routine musicali sono situate nei DATA dalle linee 10100 in poi. Se a qualcuno non dovessero piacere i motivi scelti, può tranquillamente sostituirli, variando le note riportate in tali DATA. Il programma non prevede le istruzioni per il gioco, in quanto quest'ultimo è diventato tanto popolare che pensiamo la maggior parte dei lettori le conoscano. Comunque, per chi ancora ne fosse all'oscuro, ecco una breve spiegazione delle regole: si gioca su una scacchiera di 8 x 8 caselle al cui centro sono deposte, alternate, due pedine di colore bianco e due di colore nero. Al turno ciascun giocatore depone una pedina del proprio colore, in modo da imprigionare almeno una pedina avversaria fra quella deposta e un'altra del proprio colore, già presente sulla scacchiera. Se



Elenco variabili utilizzate

A%(99)	Matrice contenente i valori delle caselle della scacchiera, in base ai quali il C16 sceglie la mossa da effettuare
B%(7)	Matrice contenente le otto direzioni possibili sulla scacchiera
C%(99)	Matrice contenente la gestione del gioco: 0 = casella libera, 16 = casella con pedina nera, 127 = casella con pedina bianca, 9 = bordo della scacchiera
D%(99)	Matrice che duplica la C%(99), indispensabile per la scelta della mossa del C16 al 2° livello di gioco
C	Contatore pedine del C16
G	Contatore pedine giocatore
PC	Valore pedine C16 (bianca = 127, nera = 16)
PG	Valore pedine del giocatore (vedasi sopra)
PP	Pedina propria
PA	Pedina avversaria
S	Mossa del giocatore
M	Variabile di controllo della validità della mossa scelta
Q	Variabile di scelta mossa C16 al 1° livello
Q2	Variabile di scelta mossa C16 al 2° livello
PM	Possibile mossa C16 al 2° livello
LV	Livello gioco

ciò non è possibile, passa il turno. Le pedine avversarie imprigionate diventano pedine del proprio colore. Il gioco termina quando tutta la scacchiera è completata oppure nessun giocatore può effettuare la mossa. Vince chi alla fine della partita ha più pedine del proprio colore.

L'inizio della gara spetta sempre al nero; pertanto, se volete far iniziare la partita al vostro C16, dovrete scegliere il colore bianco. Per quanto riguarda la scelta della mossa il computer risponde in circa 15 secondi al 1° livello di gioco ed in un tempo variabile dai 30 ai 120 secondi al 2° livello, in base alle eventuali posizioni da analizzare.

Si noti, infine, che i valori che identificano le pedine sono rispettivamente "16"

LOAD



Othello per C16

per il nero e "127" per il bianco. Questi valori sono stati intenzionalmente scelti, in quanto rappresentano i codici di tali colori (per il bianco alla sua luminosità

massima) e che, inseriti con opportune istruzioni POKE nella matrice di controllo colore video, colorano le pedine alle varie mosse eseguite.

Othello versione per C16

```
102 REM VARIABILI INIZIALI
103 REM
104 REM
110 DIMA%(99),B%(7),C%(99),D%(99)
120 FORI=0TO7:READB%(I):NEXT
130 FORI=1TO4:FORJ=1TO8:READX=Y:I*10+J
140 A%(Y)=X:A%((4-I)*20+Y+10)=X:NEXT:NEXT

150 M$="{HOME}{ 22 GIU' }":DF=3172:DC=2148

160 N$="{ 37 SPAZI }"
170 VOL8
200 REM
201 REM INIZIO GIOCO
202 REM
203 REM
204 REM
210 FORI=0TO99:C%(I)=0:NEXT
220 FORI=0TO9:C%(I)=9:C%(I+90)=9:NEXT
230 FORI=1TO8:C%(I*10)=9:C%(I*10+9)=9:NEXT
240 C%(44)=16:C%(45)=127:C%(54)=127:C%(55)=16:G=2:C=2:PG=16:PC=16
250 COLOR0,6,5:COLOR4,6,5
260 PRINT"{CLR}CON QUALE COLORE GIOCHI? (B/N)"
270 GETZ$:IFZ$<>"B"ANDZ$<>"N"THEN270
280 IFZ$="B"THENPG=127:ELSEPC=127
290 PRINT"LIVELLO GIOCO? (1/2)"
295 GETZ$:IFZ$<>"1"ANDZ$<>"2"THEN295
296 LV=ASC(Z$)-48
300 PRINT"{CLR}";TAB(5)"O T H E L L O";TAB(44)"{ 15 E}"
310 PRINTTAB(22)"1 2 3 4 5 6 7 8",TAB(21)"[<A>]C[<R>]C[<R>]C[<R>]C[<R>]C[<R>]C[<R>]C[<R>]C[<S>]"
320 FORI=1TO7:PRINTTAB(20)CHR$(64+I);"- - - - -";CHR$(64+I)
330 PRINTTAB(21)"[<Q>]C+C+C+C+C+C+C+C[<W>":NEXT
340 PRINTTAB(20)"H- - - - -H",TAB(21)"[<Z>]C[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C[<E>]C[<X>]"
350 PRINTTAB(22)"1 2 3 4 5 6 7 8":POKE350,81:POKE3502,81
360 POKE3580,81:POKE3582,81:POKE2478,127:POKE2556,127
370 PRINT"{HOME}{ 5 GIU' }";" C16 Q =";C,TAB(201)"TU{ 2 SPAZI }Q =";G:POKE2253,PC:POKE2453,PG
380 PRINTTAB(241)"LIVELLO: ";LV
```

```
900 IFPG=127THEN2000
1000 REM
1001 REM
1002 REM MOSSA DEL GIOCATORE
1003 REM
1005 RESTORE(10100)
1006 READX,Y:IFX=0THEN1005
1007 SOUND1,X,Y
1010 PRINTM$;"INSERISCI LA TUA MOSSA (LET/NUM){ 5 SPAZI}"
1020 GETZ$:IFZ$<"A"ORZ$>"H"THEN1006
1030 S=(ASC(Z$)-64)*10
1035 READX,Y:IFX=0THENRESTORE(10100):GOTO1035
1036 SOUND1,X,Y
1040 GETZ$:IFZ$<"1"ORZ$>"8"THEN1035
1050 S=S+ASC(Z$)-48:PA=PC:PP=PG:M=0
1070 PRINTM$;N$:GOSUB7000:IFM<>0THEN1100
1080 PRINTM$;"MOSSA NON PERMESSA - RIPETI"
1090 FORI=1000TO700STEP-25:SOUND1,S,5:NEXT
T:GOSUB6900:GOTO1005
1100 P=S:N=1:GOSUB6800:GOSUB6900:GOSUB6000
1110 C=C-N+1:G=G+N:IFC+G=64THEN4000
1120 GOSUB6700
2000 REM
2001 REM
2002 REM MOSSA C16
2003 REM
2004 REM
2010 PP=PC:PA=PG:PRINTM$;"PREMI UN TASTO PER LA MIA MOSSA"
2015 FORI=650TO850STEP50:SOUND1,I,5:NEXT
2020 GETZ$:IFZ$=" "THEN2020
2030 PRINTM$;N$:ONLVGOSUB2500,3000
2040 IFQ<>0THEN2100
2050 PRINTM$;"PASSO IL TURNO{ 22 SPAZI}"
2060 FORI=1000TO700STEP-25:SOUND1,I,5:NEXT
T:GOSUB6900:GOTO2200
2100 PRINTM$;"LA MIA MOSSA E' ";
2110 PRINTCHR$(INT(Q/10)+64);CHR$(Q-INT(Q/10)*10+48)
2115 FORI=700TO950STEP50:SOUND1,I,5:NEXT
2120 S=Q:P=Q:N=1:GOSUB6800:GOSUB6900:GOSUB6000
2130 C=C+N:G=G-N+1:GOSUB6700:IFC+G=64THEN4000
2200 REM
2201 REM
2202 REM CONTROLLA SE IL GIOCATORE
2203 REM PUO' MUOVERE
2204 REM
```

```

2205 REM
2210 M=0:PP=PG:PA=PC:FOR=11TO88:GOSUB700
0:IFM<>0THEN2400
2215 NEXT:IFQ=0THEN4000
2220 PRINTM$"PASSI IL TURNO":SOUND1,516,6
0
2230 FORI=1TO2:GOSUB6900:NEXT:GOTO 2000
2400 GOTO1000
2500 REM
2501 REM
2502 REM SUBROUTINE SCELTA MOSSA
2503 REM C16 AL 1 LIVELLO
2504 REM
2505 REM
2520 Q=0:FOR=11TO88:IFC%(S)<>0THEN2590
2530 IFA%(S)<A%(Q) THEN2590
2560 M=0:GOSUB7010
2570 IFA%(M)=A%(Q) ANDRND(0)<.5THENQ=M
2580 IFA%(M)>A%(Q) THENQ=M
2590 NEXT:RETURN
3000 REM
3001 REM
3002 REM SUBROUTINE SCELTA MOSSA
3003 REM C16 AL 2 LIVELLO
3004 REM
3005 REM
3010 FORI=11TO88:D%(I)=C%(I):NEXT
3020 Q2=-9:PM=0:FORK=11TO88:IFC%(K)<>0THE
N3130
3030 S=K:M=0:GOSUB7010:IFM=0THEN3130
3050 FORI=0TO7:P=M+B%(I):IFC%(P)<>PATHEN3
100
3060 P=P+B%(I):IFC%(P)=PATHEN3060
3070 IFC%(P)<>PPTHEN3100
3080 P=P-B%(I):C%(P)=PP:IFP=MTHEN3100
3090 GOTO3080
3100 NEXT:M1=M:PA=PC:PP=PG:Q=0:GOSUB2500
3105 IFQ=0THEN3140
3110 M=A%(M1)-A%(Q):IFM>Q2THENQ2=M:PM=M1
3120 FORI=11TO88:C%(I)=D%(I):NEXT
3130 PP=PC:PA=PG:NEXT:Q=PM:RETURN
3140 FORI=11TO88:C%(I)=D%(I):NEXT
3150 PP=PC:PA=PG:Q=M1:RETURN
4000 REM
4001 REM
4002 REM FINE PARTITA
4003 REM
4004 REM
4020 IFQ<0THENPRINTM$;" BRAVO - HAI VINTO
"N$
4030 IFC<0THENPRINTM$;" STAVOLTA SIAMO PA
RI "N$
4040 IFC<0THENPRINTM$;" H O { 4 SPAZI}V I
N T O !!! "N$
4050 FORI=1TO2:RESTORE(10200):DO:READX,Y:
SOUND1,X,Y
4060 LOOPUNTILX=0:NEXT:GOSUB6900
4070 PRINTM$"GIOCHI ANCORA? (S/N)"N$
4080 GETZ$:IFZ$="S"THEN200
4090 IFZ$<>"N"THEN4080
4100 PRIN"C I A O !":END
4500 PRINTM$;"{RVS}H O { 4 SPAZI}V I N T O
!!!{OFF}"
6000 REM
6001 REM
6002 REM AGGIORNAMENTO SCACCHIERA
6003 REM
6004 REM
6010 FORI=0TO7:P=S+B%(I):IFC%(P)<>PATHEN6
100
6020 P=P+B%(I):IFC%(P)=PATHEN6020
6030 IFC%(P)<>PPTHEN6100
6040 P=P-B%(I):IFP=STHEN6100

```

```

6050 GOSUB6800:C%(P)=PP:N=N+1:GOTO6040
6100 NEXT:C%(S)=PP:RETURN
6700 PRINT"{HOME}{ 5 GIU'}{ 8 DES}
{ 3 SPAZI}{ 3 SIN}"C:PRINT"{HOME}
{ 10 GIU'}{ 8 DES}{ 3 SPAZI}
{ 3 SIN}"G:RETURN
6800 D=INT(P/10)*80+2*(P-INT(P/10)*10)
6810 POKE(D+D),81:POKE(DC+D),PP:RETURN
6900 FORI=1TO1000:NEXT:RETURN
7000 REM
7001 REM
7002 REM SUBROUTINE CONTROLLO
7003 REM VALIDITA MOSSA
7004 REM
7005 REM
7006 IFC%(S)<>0THENRETURN
7010 FORI=0TO7:P=S+B%(I):IFC%(P)<>PATHEN7
050
7020 P=P+B%(I):IFC%(P)=PATHEN7020
7030 IFC%(P)<>PPTHEN7050
7040 M=S:RETURN
7050 NEXT:RETURN
10000 REM
10001 REM
10002 REM DATA
10003 REM
10004 REM
10010 DATA -11,-10,-9,-1,1,9,10,11
10020 DATA 9,2,8,6,6,8,2,9
10030 DATA 2,1,3,4,4,3,1,2
10040 DATA 8,3,7,5,5,7,3,8
10050 DATA 6,4,5,0,0,5,4,6
10100 DATA 834,14,798,14,834,28,1015,1,834
,14,798,14,834,28
10110 DATA 854,14,834,14,810,14,798,14,770
,14,798,14,810,14
10120 DATA 798,7,810,7,834,14,739,14,1015,
1,739,28,1015,1
10130 DATA 739,7,770,7,798,7,810,7,834,28
10140 DATA 854,14,770,14,1015,1,770,14,810
,14,798,14,770,14,739,28,0,0
10200 DATA 881,16,854,16,810,16,1015,1,810
,8,1015,1,810,16,739,16,854,16,834,
32
10210 DATA 739,16,854,16,834,32,739,16,834
,16,810,32
10220 DATA 881,16,854,16,810,16,1015,1,810
,8,1015,1,810,8
10230 DATA 739,16,854,16,834,32,739,16,854
,16,834,16,739,16,810,40,810,8,0,0

```





Indice di cassetta

di P.N. Peasley

trad. e adatt.

di F. Sarcino

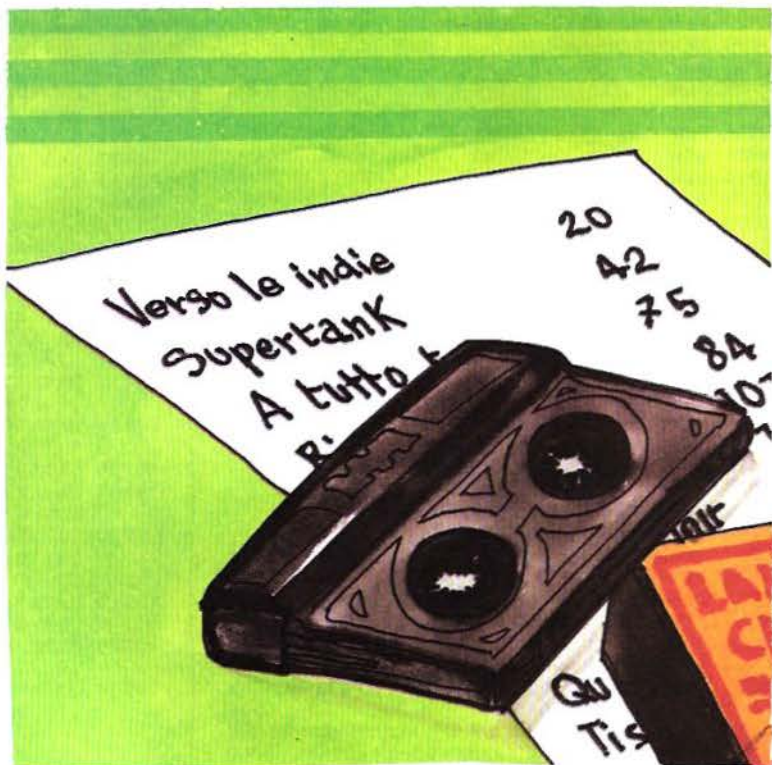
Se usate il registratore, ecco un breve programma che può aiutarvi ad organizzare le vostre cassette, creando un Indice inserito direttamente all'inizio del nastro. Per C64 e VIC 20.

Se siete stanchi di accumulare pezzetti di carta contenenti gli elenchi dei programmi registrati su ogni nastro, "Indice di Cassetta" vi offre una valida alternativa. Questo programma crea un indice, che viene registrato all'inizio di ogni facciata di ciascuna cassetta e che contiene i nomi dei programmi e la loro posizione.

Fino a 30 programmi per nastro

Se avete dato un'occhiata al listato, avrete notato che è molto breve. Tuttavia, prima di iniziare a digitarlo vi consigliamo di leggere le istruzioni contenute in questo articolo, in modo da sapere come si usa il programma. C'è da compiere anche un piccolo lavoro supplementare, ma vedrete che ne vale la pena. Ora potete cominciare a battere il listato sul vostro C64 o sul vostro VIC (con o senza espansione); fermatevi, però, quando arrivate alle istruzioni DATA (a partire dalla linea 39). Queste istruzioni dovranno in futuro contenere i nomi e le posizioni (date dalla lettura del contatore sul registratore) dei programmi. Potete notare che i nomi (PRG 01, PRG 02, ecc.) e le posizioni (000) compaiono a coppie separate da una virgola e che le parole NOME, POSIZ., sono usate per separare ogni insieme di otto coppie. Potete catalogare fino a 30 programmi su una facciata di cassetta.

Continuate adesso a digitare il programma fino in fondo e salvatelo all'inizio di una cassetta vuota. Battete RUN, correggete gli eventuali errori di battitura e risalvate il programma (sempre all'inizio del nastro). D'ora in poi chiameremo questa cassetta "nastro-destinazione", in quanto sarà quello su cui salverete i vostri programmi; le cassette da cui li caricherete le chiameremo "nastro-sorgenti".



Dato che dovrete riversare i programmi da un nastro a un altro, vi conviene riflettere su come organizzare le nuove cassette. Un paio di consigli: prima di tutto è conveniente separare i programmi di gioco, di utilità, di grafica e di suono, suddividendo eventualmente queste categorie in ulteriori sottoinsiemi. Inoltre è buona norma registrare nella prima parte delle cassette (usate al massimo delle C60!) i programmi di uso più frequente.

Una volta che avete deciso quali programmi trasferire sul nastro-destinazione (che deve essere posizionato alla fine dell'Indice di Cassetta) scrivete su un foglio di carta i loro nomi nell'ordine in cui li volete copiare.

A questo punto inserite nel registratore

il nastro-sorgente e caricate il primo programma da riversare. Inserite quindi il nastro-destinazione, azzerate il contatore ed effettuate il SAVE.

Scrivete ora accanto al nome del programma la sua lunghezza (cioè il numero segnato dal contatore dopo il salvataggio). Ripetete questa procedura per tutti i programmi che volete registrare sul nastro-destinazione, facendo attenzione ad azzerare il contatore dopo ogni SAVE.

Terminata questa operazione (la più lunga e noiosa) dovete aggiungere sul foglio una seconda colonna di numeri accanto a quella contenente la lunghezza dei programmi: questi nuovi numeri rappresentano la posizione di ogni programma. Essi vanno calcolati nel modo

seguente: segnate 000 accanto al primo nome della lista, e per ogni programma successivo calcolate la sua posizione, sommando lunghezza e posizione del precedente. Vediamo un esempio:

NOME	LUNGH.	POSIZ.
Speedscript	035	000
Sintetizzatore	012	035
Disegnatore	027	047
.....	...	...

Ora riavvolgete il nastro-destinazione, caricate l'Indice di Cassetta e listate le istruzioni DATA (battete LIST 39-). Sostituite a PRG 01, PRG 02, ecc. (utilizzando il tasto INST/DEL per cancellare o inserire i caratteri necessari), e ai corrispondenti 000, i nomi e le posizioni dei programmi registrati sulla cassetta, ricopiandoli dal vostro foglio. Completata questa operazione riavvolgete il nastro-destinazione e risalvate l'Indice di Cassetta modificato. Ora battendo RUN potrete osservare sullo schermo l'elenco dei programmi pre-

sentiti sulla cassetta e le loro rispettive posizioni.

Aggiornamento

Per aggiornare l'Indice quando dovete registrare un nuovo programma sulla cassetta seguite la seguente procedura:

1. Inserite il nastro-destinazione e caricate l'Indice di Cassetta.
2. Azzerate il contatore.
3. Leggete dall'Indice il nome dell'ultimo programma salvato e la sua posizione.
4. Fate avanzare velocemente il nastro fino a tale posizione.
5. Verificate il programma; con questa tecnica il nastro viene posizionato alla fine del programma (ignorare il messaggio ?VERIFY ERROR).
6. Prendete nota del numero indicato dal contatore.
7. Rimuovete il nastro-destinazione senza riavvolgerlo.
8. Inserite il nastro-sorgente e caricate il programma da trasferire.



9. Rimuovete il nastro-sorgente.
10. Inserite il nastro-destinazione senza riavvolgerlo.
11. Salvate il programma.
12. Riavvolgete il nastro-destinazione.
13. Caricate l'Indice di Cassetta.
14. Listate le istruzioni DATA.
15. Sostituite al posto giusto il nome del programma salvato.
16. Sostituite allo 000 successivo la posizione (il numero che avevate annotato al punto 6).
17. Riavvolgete il nastro-destinazione.
18. Salvate l'Indice di Cassetta modificato.

Indice di cassetta per VIC 20 e C64

```

1 SYS65517 :rem 56
2 IFPEEK(781)=40THENMT$="{WHT}":GOTO4 :rem 153
3 MT$="{BLU}" :rem 152
4 PRINTMT$"{CLR}{ 10 GIU' }{ 3 SPAZI }{RVS} :rem 29
  INDICE{ 2 SPAZI }CASSETTA"
5 FORQ=1TO2000:NEXT :rem 185
6 PRINT"{CLR}{ 2 GIU' }{ 3 SPAZI }{RVS}INDI :rem 174
  CE CASSETTA":PRINT"{GIU'}PER USARE L'IN
  DICE:"
7 PRINT"{GIU'} 1-AZZERA IL" :rem 217
8 PRINT"{ 3 SPAZI }CONTATORE" :rem 184
9 PRINT"{GIU'} 2-TROVA NELL'ELENCO" :rem 14
10 PRINT"{ 3 SPAZI }IL PROGRAMMA E LA" :rem 63
11 PRINT"{ 3 SPAZI }SUA POSIZIONE" :rem 220
12 PRINT"{GIU'} 3-FAI AVANZARE IL" :rem 98
13 PRINT"{ 3 SPAZI }NASTRO FINO AL" :rem 197
14 PRINT"{ 3 SPAZI }PUNTO ESATTO":PRINT" :rem 216
  { 3 SPAZI }USANDO {RVS}F.FWD"
15 PRINT"{GIU'} 4-FERMA IL NASTRO ED" :rem 9
16 PRINT"{ 3 SPAZI }ESEGUI IL {RVS}LOAD":P :rem 154
  RINT"{GIU'}{ 3 SPAZI }{RVS}PREMI UN TAS
  TO"
17 GETAN$:IFAN$=""THEN17 :rem 143
18 PRINT"{CLR}{ 2 GIU' }{ 3 SPAZI }{RVS}IND :rem 217
  ICE CASSETTA":PRINT"{GIU'}QUESTO NASTR
  O"
19 PRINT"CONTIENE I SEGUENTI" :rem 61
20 PRINT"PROGRAMMI NELLA":PRINT"POSIZIONE :rem 160
  INDICATA:{GIU'}"
21 READA$:READBS :rem 160
22 PRINTA$TAB(15)B$:PRINT :rem 17

```

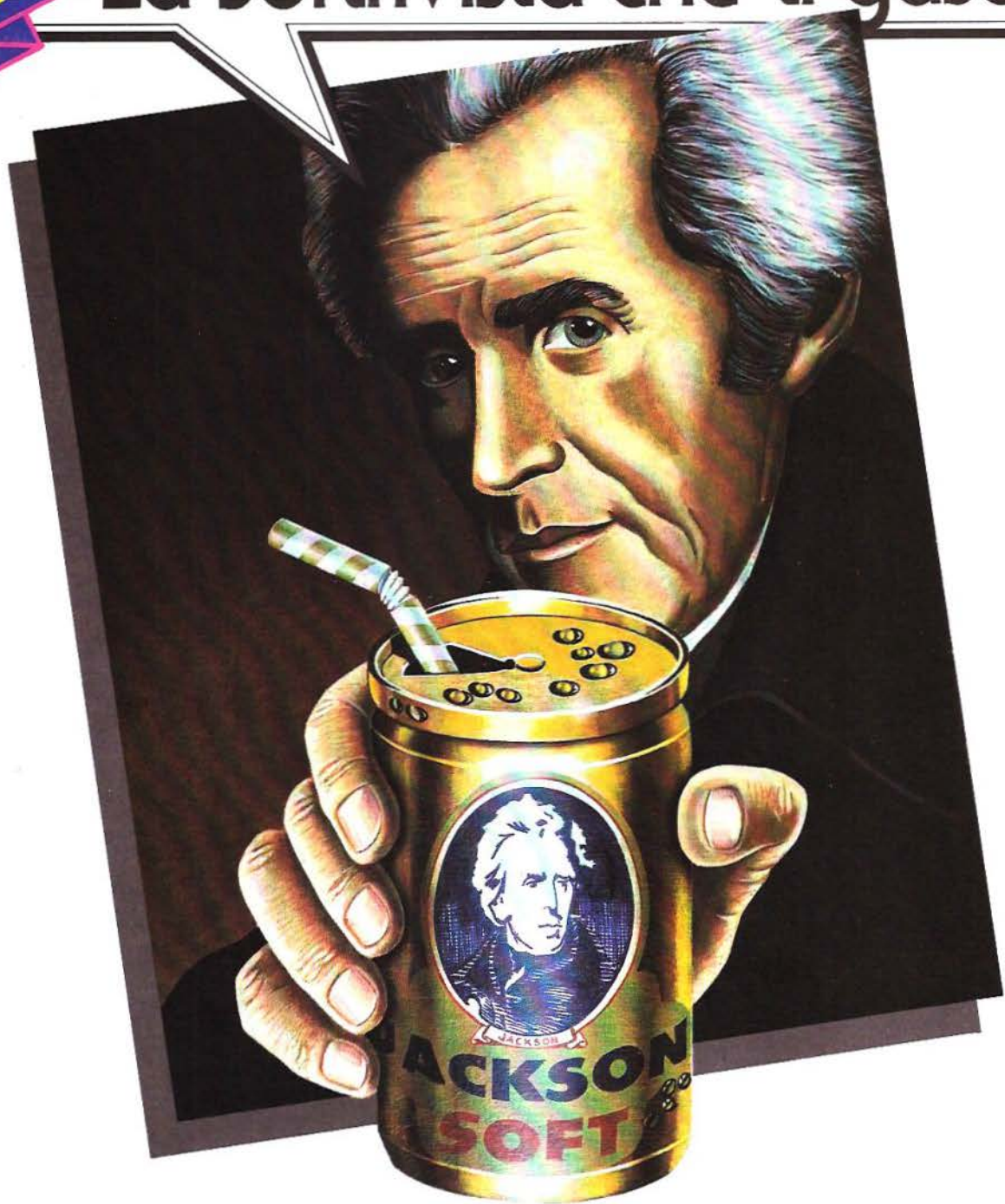
```

23 FORX=1TO8 :rem 237
24 READA$ :rem 231
25 IFAS$="-1"THEN33 :rem 207
26 READBS :rem 234
27 IFB$="-1"THEN33 :rem 210
28 PRINTA$TAB(17)B$ :rem 82
29 NEXT :rem 170
30 PRINT"{GIU'}{ 3 SPAZI }{RVS}PREMI UN TA :rem 2
  STO"
31 GETP$:IFP$=""THEN31 :rem 9
32 IFAS$<>"-1"THEN18 :rem 13
33 PRINT"{GIU'}{RVS}*** FINE{ 2 SPAZI }ELE :rem 243
  NCO ***MT$
34 PRINT"{HOME}{ 20 GIU' }ANCORA {S/N}";I :rem 44
  NPUTAN$
35 IFAN$="S"THEN38 :rem 24
36 IFAN$<>"N"THEN34 :rem 77
37 END :rem 65
38 RESTORE:GOTO18 :rem 107
39 DATANOME,POSIZ. :rem 164
40 DATAPRG 01,000,PRG 02,000,PRG 03,000 :rem 235
41 DATAPRG 04,000,PRG 05,000,PRG 06,000 :rem 245
42 DATAPRG 07,000,PRG 08,000 :rem 197
43 DATANOME,POSIZ. :rem 159
44 DATAPRG 09,000,PRG 10,000,PRG 11,000 :rem 245
45 DATAPRG 12,000,PRG 13,000,PRG 14,000 :rem 246
46 DATAPRG 15,000,PRG 16,000 :rem 199
47 DATANOME,POSIZ. :rem 163
48 DATAPRG 17,000,PRG 18,000,PRG 19,000 :rem 8
49 DATAPRG 20,000,PRG 21,000,PRG 22,000 :rem 247
50 DATAPRG 23,000,PRG 24,000 :rem 192
51 DATANOME,POSIZ. :rem 158
52 DATAPRG 25,000,PRG 26,000,PRG 27,000 :rem 0
53 DATAPRG 28,000,PRG 29,000,PRG 30,000,- :rem 139
  1

```

FINALMENTE!

La Softrivista che ti gasa!



**QUALCOSA DI SUPER, DI INEDITO,
DI IRRESISTIBILE**

IL VERO GIOCO COMINCIA ADESSO

È IN EDICOLA JACKSON SOFT SERIE ORO

La softtrivista con i giochi esclusivi per Commodore 64 e Spectrum 48K importati dall'Inghilterra, mai presentati in Italia. Una sfida Jackson al già visto, al già fatto, al... già registrato.

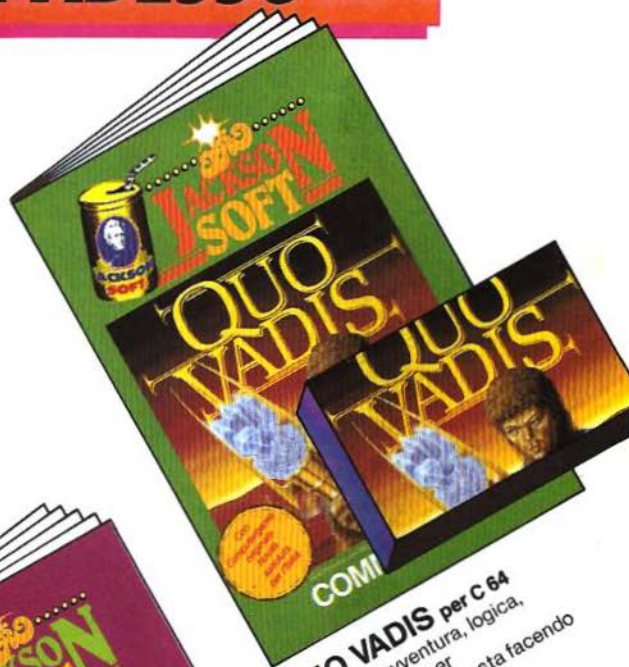


Ogni mese in edicola

- la softtrivista con:
 - 6 pagine di commento al supergame presentato;
 - 4 pagine di rubriche;
 - 22 pagine di listati scelti tra il meglio di quanto pubblicato in fatto di grafica, giochi, utility.
- una cassetta originale con il SUPERGAME del mese

Tutto a sole 10.000 lire

Corri in edicola, il vero gioco comincia solo adesso e se sei davvero bravo partecipa alla "sfida al campione", utilizzando il tagliando che troverai sull'ultima pagina di copertina di ogni numero.



QUO VADIS per C 64
Strategia, avventura, logica, inganno nel super computer game che sta facendo impazzire l'Inghilterra. Nessuno è ancora riuscito a raggiungere la fine del labirinto. Prova tu!



BRIAN BLOODAXE per Spectrum 48K
Un plat forme game che ti condurrà in pieno 21° secolo, alla conquista dei favolosi gioielli della corona inglese. Forza, astuzia, magia e... la colonna sonora di Monty Python.



**GRUPPO
EDITORIALE
JACKSON**
SAN FRANCISCO - MONZA - MILANO



Stampare lo schermo ad alta risoluzione

di **G. Peele**
trad. e adatt.
di **F. Sarcina**

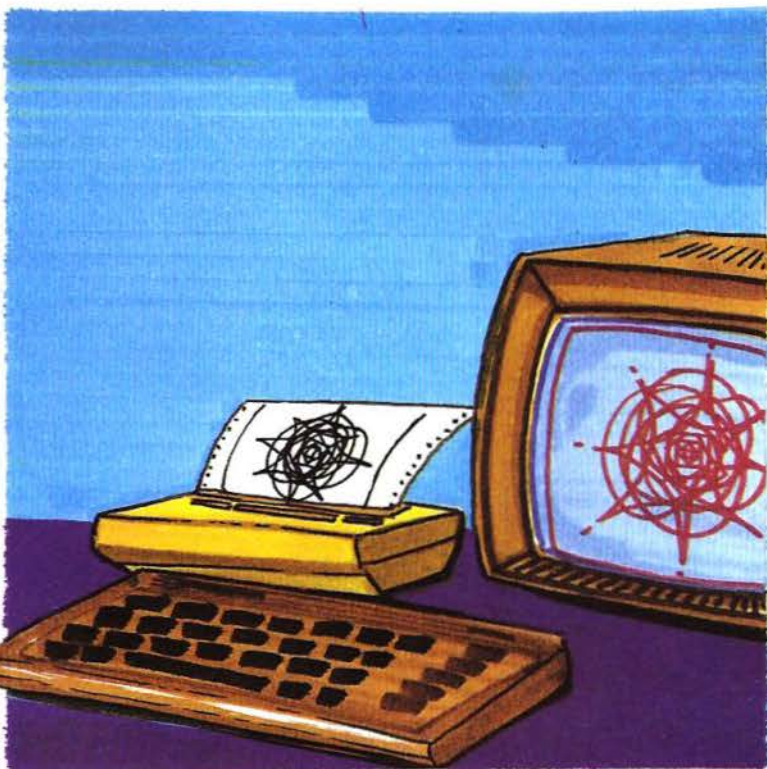
Vi è mai capitato di creare un disegno o un grafico ad alta risoluzione e desiderare di riprodurlo su carta con la vostra stampante? Questo programma vi permette di farlo. Ne presentiamo due versioni, una per il C64 e una per il VIC; quest'ultima richiede la cartuccia Super Expander. Entrambe le versioni sono compatibili con le stampanti Commodore 1525 (ma non 1526), MPS-801 e Seikosha GP-100VC.

Sia il VIC 20 che il C64 vi consentono di creare immagini ad alta risoluzione sullo schermo del vostro televisore. È facile fare disegni "artistici" con le cartucce Super Expander per VIC o 64, oppure con altri programmi di grafica. Tuttavia la maggior parte di questo software non possiede un'opzione che vi permette di riprodurre su carta i vostri capolavori. A meno che vogliate lasciare acceso il computer per un tempo indefinito, rinunciando anche ad usarlo per altri scopi, le vostre creazioni avranno breve vita!

Entrambe le versioni di questo programma funzionano con la stampante Commodore 1525 od altra compatibile, ad eccezione della 1526 e della MPS 802.

Trasferimento dei bit

Il programma di stampa dello schermo ad alta risoluzione, che d'ora in poi chiameremo per brevità schermo hi-res, è progettato per trasferire alla stampante l'informazione contenuta nei bit della memoria di schermo. La 1525, quando si trova in modo grafico, può accettare solo sette bit di informazioni per volta (il bit più alto del byte deve sempre essere settato, cioè valere 1), mentre la memoria di schermo è composta ovviamente da byte di otto bit. Ciò significa che, prima di poterli mandare alla stampante, questi byte devono essere "spezzettati" nei singoli bit. Il trasferimento dell'informazione è ulteriormente complicato dal fatto che nel C64 lo schermo hi-res può essere situato in diverse aree di memoria.



Il programma legge i dati della memoria un bit per volta, iniziando dall'angolo in basso a sinistra dello schermo (ogni bit corrisponde a un pixel, cioè ad un punto dello schermo). Raccolti sette bit, questi vanno a formare un byte di informazione, che viene inviato alla stampante. Si passa quindi al bit più a sinistra della riga superiore e così via su per lo schermo. Una volta letti i sette bit dell'angolo in alto a sinistra il programma torna in basso e legge l'ottavo bit da sinistra, il nono e via di seguito fino ad accumulare altri sette bit da inviare alla stampante. Si prosegue così ciclicamente dal basso all'alto fino a completare la lettura dello schermo. Dato che la stampa avviene da sinistra verso destra, mentre la lettura

è effettuata dal basso verso l'alto, l'immagine sulla carta risulta ruotata di 90 gradi rispetto a quella sullo schermo. Entrambe le versioni del programma sono scritte in linguaggio macchina, anche se il listato è in BASIC; le prime linee memorizzano la routine, che si trova sotto forma di istruzioni DATA, in una area libera della RAM. La parte in BASIC consente anche di scegliere se effettuare la stampa in larghezza normale o doppia, e nella versione del VIC protegge il linguaggio macchina da un'eventuale cancellazione da parte di programmi BASIC. Per un corretto funzionamento raccomandiamo di caricare e far girare questo programma prima di caricare il software che crea l'immagine

in alta risoluzione.

Stampa e selezione della larghezza

La stampa dello schermo hi-res viene comandata con SYS 3584 (per il VIC) oppure SYS 52224 (per il 64). La SYS può essere battuta direttamente per stampare il disegno correntemente sullo schermo, oppure può essere inserita in un programma che crei un disegno. Ricordatevi però che il linguaggio macchina deve venir memorizzato prima dell'esecuzione della SYS e che la stampante deve essere accesa e collegata al computer.

In entrambe le versioni la larghezza della stampa può essere selezionata, oltre che all'inizio tramite la parte BASIC del programma, anche in seguito per mezzo di un'istruzione POKE. Per l'esattezza con POKE 2, 1 si ottiene la larghezza normale e con POKE 2, 2 quella doppia.

Note per il VIC

La versione del VIC è progettata per venire usata con la cartuccia Super Expander. Il comando GRAPHIC 2 di questa espansione crea uno schermo hi-res di 3200 byte, a partire dalla locazione 7680 (\$1E00). Con i 3Kbyte aggiuntivi di RAM forniti dalla Super Expander la memorizzazione dei programmi BASIC inizia dalla locazione

1024 (\$0400). La linea 5 del programma riserva due pagine (512 byte) di memoria per il linguaggio macchina, subito prima dell'area utilizzata dallo schermo hi-res. Ciò significa che il linguaggio macchina inizia dalla locazione 3584 (\$0E00), e che l'area 1024-3583 (\$0400-\$0DFF) è disponibile per i programmi BASIC. Questi 2.5Kbyte non sono certo molti, ma possono aumentare se avete la possibilità di collegare al vostro VIC, assieme alla Super Expander, anche un'espansione RAM da 8Kbyte o più. In tal caso i 3Kbyte di RAM della Super Expander non vengono più usati per il BASIC, ma possono continuare a contenere la routine in linguaggio macchina, che viene sempre chiamata con SYS 3584. Tuttavia in questo caso è necessario rimuovere dal programma la linea 5, che fissa a 3583 (\$0DFF) il limite della memoria BASIC; si otterrà altrimenti il messaggio: OUT OF MEMORY ERROR. In questa configurazione lo schermo hi-res creato dal comando GRAPHIC 2 occupa i 3200 byte a partire dalla locazione 4096 (\$1000), e la RAM disponibile per il BASIC inizia da 8192 (\$2000) e prosegue fino alla fine dell'espansione di memoria collegata.

Note per il C64

Nella versione del C64 il linguaggio macchina risiede nell'ultima parte del blocco di RAM che inizia dalla locazione

49152 (\$C000), e quindi non interferisce con la memoria BASIC, che occupa invece l'area 2048-40959 (\$0800-\$9FFF). Ciò rende la routine compatibile con la Super Expander 64, ma non col programma DOS wedge di gestione del disk drive, che va ad occupare proprio la stessa area.

Questa routine è progettata per stampare lo schermo hi-res mentre esso appare sul video; in pratica l'istruzione SYS 52224 va inserita in un programma che crea un disegno ad alta risoluzione e lo mostra sul video. È comunque possibile effettuare la stampa anche quando lo schermo è a bassa risoluzione (cioè quando vengono visualizzati numeri e lettere e compare il cursore): è il caso che si presenta se volete battere la SYS direttamente da tastiera. In tal caso, però, è prima necessario effettuare una POKE nella locazione 900, assegnando ad essa il valore del byte alto dell'indirizzo iniziale dello schermo hi-res. La stampa viene poi comandata tramite una SYS alla locazione 52224+32, cioè 52256. Questa nuova SYS salta la parte iniziale della routine, che serve a determinare l'area di memoria occupata dello schermo hi-res.

Facciamo un esempio per chiarire le idee. Se il vostro schermo hi-res inizia dalla locazione 57344 (\$E000), per stampare ciò che vi è disegnato battete:

POKE 900, 57344/256 : SYS 52256

Programma per VIC 20

```
5 POKE56,14:CLR :rem 123
6 INPUT"[CLR]{BLU}LARGHEZZA (1-2)";W$:IF
  VAL(W$)<1ORVAL(W$)>2THEN6 :rem 224
7 POKE2,VAL(W$) :rem 188
10 I=3584:PRINT"[CLR]MEMORIZZAZIONE":PRIN
  T"LINGUAGGIO MACCHINA" :rem 193
20 READA:IFA=256THEN40 :rem 54
30 POKEI,A:I=I+1:CH=CH+A:GOTO20 :rem 123
40 IFCH<>"35292"THENPRINT"[GIU']ERRORE NELL
  E":PRINT"ISTRUZIONI DATA":END:rem 163
50 PRINT"[GIU']SYS 3584 PER INIZIARE LA C
  OPIA":NEW :rem 121
3584 DATA169,16,141,132,3,169,0 :rem 44
3591 DATA32,189,255,169,4,170,160 :rem 154
3598 DATA255,32,186,255,32,192,255 :rem 212
3605 DATA162,4,32,201,255,176,3 :rem 34
3612 DATA76,34,14,76,5,15,169 :rem 207
3619 DATA8,32,210,255,169,13,32 :rem 42
3626 DATA210,255,162,0,169,1,141 :rem 82
3633 DATA104,15,169,0,141,105,15 :rem 78
3640 DATA169,0,141,106,15,169,159 :rem 145
3647 DATA141,107,15,32,225,255,208 :rem 190
3654 DATA3,76,5,15,138,72,152 :rem 204
3661 DATA72,32,14,15,104,168,104 :rem 84
3668 DATA170,173,111,15,45,109,15 :rem 142
3675 DATA240,12,173,108,15,13,104 :rem 131
3682 DATA15,141,108,15,76,116,14 :rem 90
```

```
3689 DATA173,104,15,73,255,45,108 :rem 154
3696 DATA15,141,108,15,14,104,15 :rem 85
3703 DATA173,104,15,201,128,240,20 :rem 171
3710 DATA24,173,105,15,105,1,141 :rem 71
3717 DATA105,15,173,106,15,105,0 :rem 78
3724 DATA141,106,15,76,66,14,173 :rem 94
3731 DATA108,15,9,128,224,22,144 :rem 88
3738 DATA10,173,108,15,41,63,9 :rem 249
3745 DATA128,141,108,15,142,113,15 :rem 183
3752 DATA166,2,142,112,15,168,32 :rem 87
3759 DATA210,255,152,206,112,15,208 :rem 237
3766 DATA246,174,113,15,169,1,141 :rem 146
3773 DATA104,15,169,0,141,108,15 :rem 86
3780 DATA56,173,105,15,233,6,141 :rem 90
3787 DATA105,15,173,106,15,233,0 :rem 87
3794 DATA141,106,15,206,107,15,173 :rem 189
3801 DATA107,15,201,255,240,3,76 :rem 81
3808 DATA66,14,224,22,176,31,24 :rem 42
3815 DATA173,105,15,105,7,141,105 :rem 131
3822 DATA15,173,106,15,105,0,141 :rem 75
3829 DATA106,15,232,169,159,141,107 :rem 249
3836 DATA15,169,13,32,210,255,76 :rem 97
3843 DATA66,14,169,13,32,210,255 :rem 93
3850 DATA32,231,255,96,173,105,15 :rem 143
3857 DATA74,74,74,168,185,62,15 :rem 69
3864 DATA133,253,185,83,15,133,254 :rem 201
```

```

3871 DATA173,105,15,41,7,73,7 :rem 201
3878 DATA168,200,169,0,56,42,136:rem 104
3885 DATA208,252,141,109,15,172,107 :rem 245
3892 DATA15,177,253,45,109,15,141 :rem 150
3899 DATA111,15,96,0,160,64,224 :rem 48
3906 DATA128,32,192,96,0,160,64 :rem 48
3913 DATA224,128,32,192,96,0,160:rem 92
3920 DATA64,224,128,16,16,17,17 :rem 43
3927 DATA18,19,19,20,21,21,22 :rem 192
3934 DATA22,23,24,24,25,26,26 :rem 193
3941 DATA27,27,28,256 :rem 72

```

Programma per C64

```

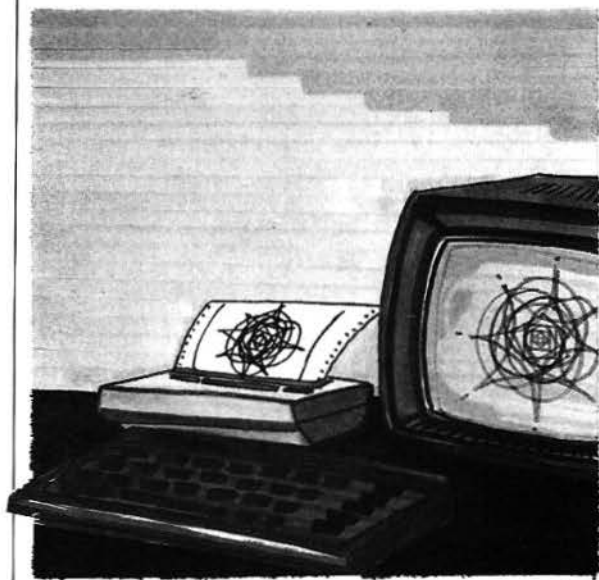
3 INPUT"{CLR}{WHT}LARGHEZZA (1-2)";WIS :rem 39
4 IFVAL(WIS)<1ORVAL(WIS)>2THEN3 :rem 147
5 POKE2,VAL(WIS) :rem 186
10 I=52224 :rem 230
20 READA:IFA=256THEN40 :rem 54
25 PRINT"[CLR]MEMORIZZAZIONE LINGUAGGIO M :rem 158
ACCHINA":POKE646,A :rem 123
30 POKEI,A:I=I+1:CH=CH+A:GOTO20 :rem 123
40 IFCH<>60660THENPRINT"ERRORE NELLE ISTR :rem 132
UZIONI DATA":END :rem 132
50 PRINT"[GIU']SYS 52224 PER INIZIARE LA :rem 164
COPIA":NEW :rem 164
52224 DATA173,0,221,41,3,73 :rem 87
52230 DATA3,160,6,10,136,208 :rem 135
52236 DATA252,141,132,3,173,24 :rem 241
52242 DATA208,41,8,240,9,24 :rem 97
52248 DATA169,32,109,132,3,141 :rem 249
52254 DATA132,3,169,0,32,189 :rem 152
52260 DATA255,169,4,170,160,255 :rem 48
52266 DATA32,186,255,32,192,255 :rem 56
52272 DATA162,4,32,201,255,176 :rem 247
52278 DATA3,76,61,204,76,32 :rem 109
52284 DATA205,169,8,32,210,255 :rem 254
52290 DATA169,13,32,210,255,162 :rem 41
52296 DATA0,169,1,141,198,205 :rem 206
52302 DATA169,0,141,199,205,169 :rem 50
52308 DATA0,141,200,205,169,199 :rem 42
52314 DATA141,201,205,32,225,255:rem 79
52320 DATA208,3,76,32,205,138 :rem 196
52326 DATA72,152,72,32,41,205 :rem 195
52332 DATA104,168,104,170,173,205 :rem 136
52338 DATA205,45,203,205,240,12 :rem 32
52344 DATA173,202,205,13,198,205:rem 91
52350 DATA141,202,205,76,143,204:rem 81
52356 DATA173,198,205,73,255,45 :rem 62
52362 DATA202,205,141,202,205,14:rem 73
52368 DATA198,205,173,198,205,201 :rem 158
52374 DATA128,240,20,24,173,199 :rem 50
52380 DATA205,105,1,141,199,205 :rem 38
52386 DATA173,200,205,105,0,141 :rem 30
52392 DATA200,205,76,93,204,173 :rem 46
52398 DATA202,205,9,128,224,45 :rem 1
52404 DATA144,10,173,202,205,41 :rem 26
52410 DATA31,9,128,141,202,205 :rem 235
52416 DATA142,207,205,166,2,142 :rem 37
52422 DATA206,205,168,32,210,255:rem 87
52428 DATA152,206,206,205,208,246 :rem 144
52434 DATA174,207,205,169,1,141 :rem 43
52440 DATA198,205,169,0,141,202 :rem 40
52446 DATA205,56,173,199,205,233:rem 106
52452 DATA6,141,199,205,173,200 :rem 43

```

```

52458 DATA205,233,0,141,200,205 :rem 28
52464 DATA206,201,205,173,201,205 :rem 130
52470 DATA201,255,240,3,76,93 :rem 201
52476 DATA204,224,45,176,31,24 :rem 253
52482 DATA173,199,205,105,7,141 :rem 51
52488 DATA199,205,173,200,205,105 :rem 149
52494 DATA0,141,200,205,232,169 :rem 36
52500 DATA199,141,201,205,169,13:rem 89
52506 DATA32,210,255,76,93,204 :rem 251
52512 DATA169,13,32,210,255,32 :rem 242
52518 DATA231,255,96,173,201,205:rem 97
52524 DATA41,7,141,204,205,173 :rem 242
52530 DATA201,205,74,74,74,168 :rem 255
52536 DATA185,146,205,133,251,185 :rem 152
52542 DATA172,205,133,252,24,165:rem 91
52548 DATA251,109,204,205,133,251 :rem 140
52554 DATA165,252,105,0,133,252 :rem 38
52560 DATA24,173,132,3,101,252 :rem 237
52566 DATA133,252,173,199,205,41:rem 104
52572 DATA7,73,7,168,200,169 :rem 164
52578 DATA0,56,42,136,208,252 :rem 207
52584 DATA141,203,205,24,173,200:rem 83
52590 DATA205,101,252,133,252,173 :rem 136
52596 DATA199,205,41,248,168,138:rem 121
52602 DATA72,120,162,52,134,1 :rem 186
52608 DATA177,251,162,55,134,1 :rem 254
52614 DATA88,168,104,170,152,45 :rem 53
52620 DATA203,205,141,205,205,96:rem 84
52626 DATA0,64,128,192,0,64 :rem 102
52632 DATA128,192,0,64,128,192 :rem 0
52638 DATA0,64,128,192,0,64 :rem 105
52644 DATA128,192,0,64,128,192 :rem 3
52650 DATA0,64,0,1,2,3 :rem 88
52656 DATA5,6,7,8,10,11 :rem 155
52662 DATA12,13,15,16,17,18 :rem 96
52668 DATA20,21,22,23,25,26 :rem 94
52674 DATA27,28,30,31,256 :rem 9

```





La statistica diventa facile

di **A. Burke Luitich**
trad. e adatt.
di **F. Sarcina**

I metodi statistici fondamentali possono aiutarci a prendere decisioni logiche nelle situazioni di tutti i giorni.

Nella maggior parte dei casi i metodi statistici elementari misurano un gruppo di cose (appartenenti a un dato insieme), per poi vedere come le misure effettuate variano rispetto ad un certo standard.

Questo programma di statistica, che pubblichiamo nelle due versioni per C64 e VIC 20, trasforma i dati grezzi che voi impostate in cifre, che potete utilizzare per prendere le decisioni di tutti i giorni, come ad esempio trovare il modo migliore per costruire un muro o sapere quanto denaro portare con voi quando andate a fare la spesa.

Come primo esempio consideriamo due modi possibili per tagliare una tavola di legno con precisione, e cioè usare una sega elettrica da tavolo oppure un seghetto a mano. Regoliamo la sega da tavolo sulla lunghezza di 10 cm. e tagliamo cinque pezzi. Altri cinque li tagliamo col seghetto, e poi misuriamo la lunghezza esatta dei dieci pezzi per controllare la precisione dei tagli.

Se non sono intervenuti elementi estranei durante le operazioni di taglio svolte, possiamo aspettarci che le lunghezze dei pezzi varino in conseguenza del procedimento usato. Un insieme di dati di questo tipo viene chiamato dagli statistici "campione casuale non influenzato".

Supponiamo che le misure siano le seguenti:

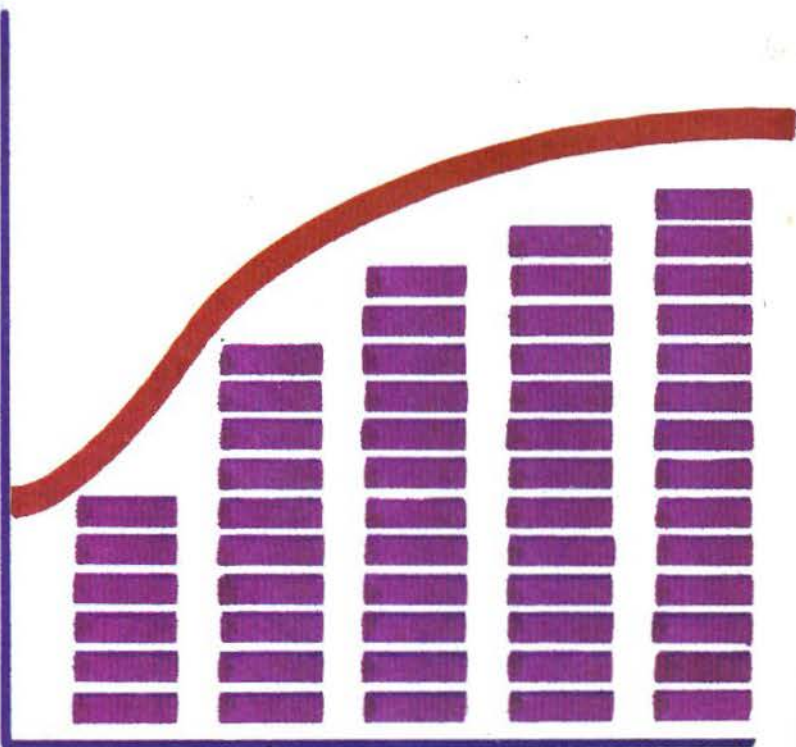
Sega da tavolo (lunghezze in cm.)	Seghetto (lunghezze in cm.)
10.5	12.2
9.8	9.1
10.3	8.0
10.7	12.8
9.6	8.8

La stessa media

Una semplice occhiata ai valori suggerì-

sce che il taglio con il seghetto è decisamente meno preciso di quello effettuato con la sega da tavolo. Tuttavia, se sommate le lunghezze relative ad ognuno dei due metodi e dividete per 5, troverete che entrambi i sistemi di taglio danno la stessa lunghezza media ("valor medio" dei campioni), e cioè 10.18 cm. Quindi il solo calcolo della lunghezza media non ci dice molto. Ciò che ci interessa conoscere è un indice dello "sparpagliamento" dei valori, in modo da determinare quale metodo ci garantisca il maggior numero di lunghezze il più vicino possibile al nostro standard di 10 centimetri. In termini statistici dobbiamo calcolare la "variazione" (o "intervallo") e la "deviazione standard".

La variazione si trova sottraendo, per ogni metodo di taglio, la lunghezza minore dalla maggiore. Per il seghetto abbiamo un intervallo di 4.8 cm. (12.8 - 8.0), mentre per la sega da tavolo la variazione è di 1.1 cm. (10.7 - 9.6). Si nota immediatamente come la sega elettrica tagli con maggior precisione: il suo intervallo è infatti più piccolo. La deviazione standard, invece, serve (assieme alla media) a prevedere con che probabilità si può presentare una certa misura. Non dovete preoccuparvi di come la si calcola; il programma lo fa per voi. Se provate ad impostare le lunghezze della tabella riguardante il seghetto, il programma vi comunicherà una deviazione standard di 2.17 cm.);



per la sega da tavolo la d.s. è invece di 0.47 cm.

Grado di precisione

Se tagliamo un gran numero di pezzi, misuriamo le loro lunghezze e le riportiamo in un grafico, ponendo in ascissa (sull'asse x) la misura in centimetri e in ordinata (sull'asse y) il numero di volte che ogni misura si presenta, otteniamo una curva a forma di campana (detta "distribuzione normale"), la cui sommità corrisponde alla misura che si è presentata più frequentemente. Combinando la deviazione standard con la lunghezza media possiamo determinare un intervallo di misure che comprende il 68.3% di tutte le lunghezze (ancora una volta non è necessario che conosciate la teoria: usate i numeri!). Per capire meglio questo concetto considerate la lunghezza media (10.18 cm.) e sottraete da essa la d.s. per il seghetto (2.17 cm.), ottenendo 8.01 centimetri. Sommate ora la d.s. alla lunghezza media e otterrete 12.35 centimetri. Il significato della deviazione standard è che nell'intervallo compreso tra 8.01 e 12.35 cm. cade il 68.3% delle lunghezze. Sommando e sottraendo la d.s. relativa alla sega da tavolo (0.47 cm.) dalla lunghezza media (10.18 cm.), troviamo che il 68.3% (oltre due terzi) delle misure cade nell'intervallo compreso tra 9.71 e 10.65 cm.

Se desiderate ottenere un campione più vasto, allora dovete aumentare il numero di deviazioni standard. Per includere il 95.4% di tutte le lunghezze dovete sommare e sottrarre dalla media due d.s., cioè il doppio della deviazione standard. Per il seghetto due d.s. corrispondono a 4.34 cm.; combinando questo dato con la lunghezza media otteniamo un intervallo che va da 5.84 a 14.52 centimetri. L'intervallo relativo alla sega elettrica va invece da 9.24 a 11.02 cm. (10.18 più e meno 0.94).

Pianifichiamo le spese

Potete usare un metodo analogo per curare la pianificazione delle vostre spese per prodotti alimentari. In questo caso i vostri dati consistono nelle somme che spendete in cibo lungo un periodo di 13 settimane (un quarto di un anno):

Settimana	Somma	Settimana	Somma
1	84000	8	94000
2	100000	9	130000
3	150000	10	98000
4	74000	11	86000
5	102000	12	104000
6	90000		
7	112000	13	108000

Se impostate questi dati nel programma, troverete che la vostra spesa media mensile è di circa 102000 lire; che la spesa varia da un minimo di 74000 a un massimo di 150000 lire, con una variazione di 76000 lire; che spendete altrettanto spesso più di 100000 lire e meno di tale somma (questo è il valore "mediano", cioè quello che divide a metà i dati); che la vostra deviazione standard è di circa 20000 lire.

Applicazioni del programma

Combinando la deviazione standard con la spesa media troviamo che in due terzi delle settimane spendete una somma compresa tra 82000 e 122000 lire in generi alimentari. In un sesto dei casi la spesa è inferiore a 82000 lire, mentre un sesto dei conti supera le 122000 lire. In tal modo, se mettete in bilancio 122000 lire per la spesa mensile, questa somma vi basterà nell'84% dei casi.

16%	68%	16%
meno di L. 82000	L. 20000 <-----> una deviaz. standard in meno	L. 20000 <-----> una deviaz. standard in più
L. 82.000	L. 102000 (media)	L. 122.000

Se volete essere sicuri di avere abbastanza denaro anche nel caso che i prezzi aumentino, allora vi conviene usare due deviazioni standard. Sommando due d.s. (40000 lire) alla spesa media (102000 lire) troverete che per avere una sicurezza del 98% dovete mettere in bilancio 142000 lire alla settimana. Naturalmente ci sono altri fattori da considerare (vacanze, feste, visite di amici o parenti), che possono alterare il vostro bilancio. Il programma non può

ovviamente tener conto di queste cose; esso è comunque un utile strumento per prendere le decisioni un po' meno a occhio e croce di quanto in genere si faccia.

Il programma chiede, prima di elaborare i risultati, di conoscere la dimensione del campione statistico, cioè il numero di dati che devono venir impostati (linea 410). Dopo l'INPUT dei valori che costituiscono il campione il programma calcola tutti i parametri statistici di cui si è parlato in questo articolo, vale a dire media, deviazione standard, mediana e variazione.

Le linee 325-350 e 4900-5610 forniscono a chi usa il programma una breve spiegazione del significato delle statistiche che vengono effettuate. La misura del campione è limitata a 100 elementi nella versione per il VIC 20, mentre il C64 può elaborare fino a 300 dati. Questi limiti sono più che sufficienti nella maggior parte dei casi.

Correzione degli errori

Una routine per la correzione degli errori è inclusa nelle linee 555-580 e 5900-6190. Essa permette la modifica di qualsiasi dato impostato, prima che vengano effettuati i calcoli. È possibile che il tempo di attesa durante l'elaborazione superi i due minuti, in quanto il programma deve ordinare i dati; ciò è normale per il BASIC, e l'attesa può anche essere più lunga se il numero di dati supera 100.

La versione per il VIC richiede un'espansione di memoria di almeno 3Kbyte. Se però vengono eliminate le spiegazioni, la routine di correzione errori e i titoli, allora il programma potrà funzionare anche su un VIC inespanso. Per la precisione, le linee da cancellare sono: 95-180, 325-350, 555-580, 4900-5610 e 5900-6190.

Ulteriori riduzioni possono essere fatte diminuendo la misura massima del campione. Per far ciò dovete cambiare il dimensionamento del vettore SA nella linea 90 e sostituire al valore 100 nella linea 420 la nuova misura massima. Una statistica su un campione di 100 valori richiede circa 30-45 minuti di calcolo manuale. Questo programma invece richiede 8-10 minuti, compreso l'INPUT.

La statistica diventa facile versione per VIC 20

```

90 DIMSA(100) :rem 185
100 PRINT "{CLR}":POKE36879,126:PRINT
    {BLK} :rem 207
120 PRINTTAB(4) "{ 8 GIU' } {RVS} LA
    { 2 SPAZI } STATISTICA" :rem 224
140 PRINTTAB(4) "{ GIU' } {RVS} DIVENTA FACILE
    " :rem 148
180 FORK=1 TO 4000: NEXT :rem 25
190 PRINT "{CLR}" :rem 254

```

```

200 PRINT "{ 2 GIU' } { 3 SPAZI } QUESTO PROGR
    AMMA" :rem 12
210 PRINT "{ 2 SPAZI } CALCOLA I SEGUENTI" :rem 0
220 PRINT " VALORI A PARTIRE DAI" :rem 88
230 PRINT "{ 3 SPAZI } DATI INTRODOTTI:" :rem 210
240 PRINT "{ 2 GIU' } 1. VALOR MEDIO" :rem 218
250 PRINT "{ GIU' } 2. DEVIATION STANDARD" :rem 24
290 PRINT "3. MEDIANA" :rem 188

```

```

300 PRINT"{GIU'}4. VARIAZIONE" :rem 207
310 PRINT"{ 3 GIU'}PREMI {RVS}C{OFF} PER
CONTINUARE" :rem 219
320 GET$=IF$<>"C"THEN320 :rem 255
330 PRINT"{CLR}":POKE214,10:PRINT:POKE211
,4:PRINT"HAI BISOGNO DI" :rem 38
340 PRINT"{GIU'}{ 2 SPAZI}ISTRUZIONI
{ 2 SPAZI}(S/N)?":GOSUB7010 :rem 123
350 IF$="S"THENGOSUB5010 :rem 238
370 PRINT"{CLR}IMPOSTA I DATI UN"
:rem 41
380 PRINT"VALORE PER VOLTA." :rem 208
410 POKE214,10:PRINT:POKE211,0:INPUT"QUAN
TI DATI";N :rem 147
420 IFN>100ORN<2THEN5710 :rem 155
430 PRINT"{CLR}{ 4 GIU'}SE COMMETTI UN ER
RORE" :rem 171
440 PRINT"{GIU'}CONTINUA A IMPOSTARE
{ 2 SPAZI}{GIU'}I DATI." :rem 122
450 PRINT"{GIU'}POTRAI FARE LE" :rem 249
460 PRINT"{GIU'}CORREZIONI PIU' TARDI."
:rem 55
470 PRINT"{ 2 GIU'}{RVS}{BLK}{ 4 DES}PREM
I UN TASTO" :rem 79
480 GET$=IF$="S"THEN480 :rem 141
500 FORI=1TON :rem 36
520 PRINT"{CLR}":POKE214,11:PRINT:POKE211
,0:PRINT"DATO N."I;:INPUTR$ :rem 28
530 SA(I)=VAL(R$) :rem 173
550 NEXT :rem 217
560 PRINT"{CLR}":POKE214,11:PRINT:POKE211
,0:PRINT"VUOI FARE DELLE" :rem 127
570 PRINT"CORREZIONI (S/N)?":GOSUB7010
:rem 83
580 IF$="S"THENPRINT"{CLR}":GOTO6000
:rem 74
587 PRINT"{CLR}":POKE214,9:PRINT:POKE211,
4:PRINT"{RVS}ATTENDI{ 2 SPAZI}PREGO"
:rem 52
588 POKE214,11:PRINT:PRINT" CALCOLO
{ 2 SPAZI}STATISTICHE" :rem 13
590 FORI=1TON :rem 45
600 SUM=SUM+SA(I) :rem 22
610 NEXT :rem 214
620 MEAN=SUM/N :rem 104
630 FORI=1TON :rem 40
640 DFF=DFF+(SA(I)-MEAN)^2 :rem 255
650 NEXT :rem 218
660 SDDEV=SQR(DFF/(N-1)) :rem 146
670 FL=0 :rem 187
680 FORI=1TON-1 :rem 139
690 IFSA(I)<=SA(I+1)THEN740 :rem 41
700 Q=SA(I) :rem 83
710 SA(I)=SA(I+1) :rem 141
720 SA(I+1)=Q :rem 177
730 FL=1 :rem 154
740 NEXT :rem 218
750 IFFL=1THEN670 :rem 247
760 RG=SA(N)-SA(1) :rem 233
765 LR=SA(1) :rem 147
767 HR=SA(N) :rem 174
810 IFN/2<>INT(N/2)THEN814 :rem 8
811 IFSA(N/2)=SA(N/2+1)THENMDD=SA(N/2)
:rem 191
812 IFSA(N/2)<>SA(N/2+1)THENMDD=(SA(N/2)+
SA(N/2+1))/2 :rem 202
813 GOTO1310 :rem 154
814 MDD=SA(INT(N/2+1)) :rem 219
1310 PRINT"{CLR}RISULTATI DEI CALCOLI"
:rem 179
1320 PRINT"*****";
:rem 110

```

```

1330 PRINT"{GIU'}NUMERO DATI"SPC(3)N
:rem 89
1340 PRINT"{GIU'}MEDIA"SPC(9)INT(MEAN*100
00+.5)/10000 :rem 160
1350 PRINT"{GIU'}DEV. ST."SPC(6)INT(SDDEV
*10000+.5)/10000 :rem 117
1360 PRINT"{GIU'}MEDIANA"SPC(7)INT(MDD*10
000+.5)/10000 :rem 227
1370 PRINT"{GIU'}VARIAZIONE"SPC(4)INT(RG*
10000+.5)/10000 :rem 174
1375 PRINT"{GIU'}VAL. + BASSO"SPC(2)LR
:rem 109
1380 PRINT"{GIU'}VAL. + ALTO"SPC(3)HR
:rem 30
1480 POKE214,21:PRINT:POKE211,4:PRINT"
{RVS}PREMI UN TASTO"; :rem 73
1490 GETA$=IFA$="S"THEN1490 :rem 191
1510 PRINT"{CLR}":POKE214,11:PRINT:POKE211
,0:PRINT"VUOI ELABORARE ALTRI"
:rem 48
1520 PRINT"DATI (S/N)?":GOSUB7010
:rem 157
1530 IF$<>"S"THENPOKE36879,27:PRINT"
{CLR}":END :rem 154
1540 CLR:DIMS A(100):GOTO330 :rem 62
5010 PRINT"{CLR}{GIU'}IL MASSIMO NUMERO D
I" :rem 76
5020 PRINT"DATI CHE PUOI" :rem 199
5030 PRINT"IMPOSTARE E' 100," :rem 118
5035 PRINT"IL MINIMO E' 2." :rem 200
5040 PRINT"{GIU'}*****"
:rem 29
5050 PRINT"{GIU'}IL VALOR MEDIO E' LA"
:rem 44
5060 PRINT"MEDIA ARITMETICA DEI" :rem 177
5070 PRINT"NUMERI IMPOSTATI." :rem 85
5090 PRINT"{GIU'}*****"
:rem 34
5100 PRINT"{GIU'}LA DEVIATIONE STANDARD";
:rem 175
5110 PRINT"E' UNA MISURA DI" :rem 70
5120 PRINT"QUANTO I NUMERI SI" :rem 38
5130 PRINT"DISCOSTANO DALLA" :rem 239
5135 PRINT"MEDIA." :rem 45
5140 PRINT"{ 2 GIU'}PREMI {RVS}C{OFF} PER
CONTINUARE"; :rem 59
5150 GETA$=IFA$<>"C"THEN5150 :rem 57
5160 PRINT"{CLR}I DATI TENDONO A"
:rem 243
5170 PRINT"CREARE UNA CURVA A" :rem 246
5180 PRINT"FORMA DI CAMPANA" :rem 146
5190 PRINT"(DISTRIBUZIONE" :rem 179
5200 PRINT"NORMALE); LA DEV. ST."
:rem 121
5210 PRINT"E' LA MISURA DELL'" :rem 171
5220 PRINT"AREA SOTTO LA CURVA." :rem 136
5230 PRINT"{GIU'}N. DI DEV.{ 5 SPAZI}% AR
EA" :rem 0
5240 PRINT"ST. (+/-)" :rem 73
5245 PRINT"-----{ 5 SPAZI}-----"
:rem 113
5250 PRINTTAB(4)"{GIU'}1"TAB(16)"68.3"
:rem 221
5260 PRINTTAB(4)"{GIU'}2"TAB(16)"95.5"
:rem 225
5270 PRINTTAB(4)"{GIU'}3"TAB(16)"99.7"
:rem 233
5280 PRINTTAB(4)"{GIU'}4"TAB(16)"99.9"
:rem 237
5290 PRINT"{ 2 GIU'}PREMI {RVS}C{OFF} PER
CONTINUARE"; :rem 65
5300 GETA$=IFA$<>"C"THEN5300 :rem 51

```



**La statistica
diventa facile**

```

5430 PRINT"{CLR}{GIU'}*****
***" :rem 179
5435 PRINT"{GIU'}LA MEDIANA E' IL" :rem 48
5440 PRINT"VALORE CHE DIVIDE A" :rem 45
5450 PRINT"META' I DATI." :rem 134
5460 PRINT"{ 2 GIU'}PREMI {RVS}C{OFF} PER
CONTINUARE"; :rem 64
5470 GETAS:IFAS<>"C"THEN5470 :rem 67
5490 PRINT"{CLR}{ 2 GIU'}LA VARIAZIONE E'
LA" :rem 214
5500 PRINT"DIFFERENZA TRA IL" :rem 245
5510 PRINT"DATO PIU' BASSO E" :rem 150
5520 PRINT"QUELLO PIU' ALTO." :rem 226
5530 PRINT"{GIU'}ESSA DA' UNA STIMA" :rem 233
5540 PRINT"APPROSSIMATIVA DELLO":rem 66
5550 PRINT"'SPARPAGLIAMENTO'" :rem 85
5555 PRINT"DEI DATI." :rem 199
5560 PRINT"{ 2 GIU'}LA DEV. STANDARD E'" :rem 26
5570 PRINT"PIU' ATTENDIBILE." :rem 10
5580 PRINT"{ 4 GIU'}PREMI {RVS}I{OFF} PER
INIZIARE" :rem 147
5600 GETAS:IFAS<>"I"THEN5600 :rem 63
5610 RETURN :rem 172
5710 POKE214,10:PRINT:POKE211,18:PRINT"
{ 4 SPAZI}" :rem 132
5720 GOTO410 :rem 156
5740 PRINT"{CLR}":POKE214,20:PRINT:POKE21
1,0:PRINT"{BLK}{RVS} {RED}C{BLK}ORRE
Z. {RED}P{BLK}ROSEG. {RED}F{BLK}INE
";RETURN :rem 11
5810 GETZ$:IFZ$="OR(Z$<>"C"ANDZ$<>"P"AND
Z$<>"F")THEN5810 :rem 22
5820 KL=0:IFZ$="P"THENKL=1:GOTO6130 :rem 92
5830 IFZ$="F"THEN587 :rem 119
5840 PRINT"{CLR}" :rem 53
6000 POKE214,11:PRINT:POKE211,0 :rem 58
6010 PRINT"RICORDI IL NUMERO DEL DATO SBA
GLIATO (S/N)?":GOSUB7010 :rem 132
6020 IFZ$="N"THEN6130 :rem 157
6030 PRINT"{CLR}":POKE214,6:PRINT:POKE211
,1:INPUT"N. DATO";EN$ :rem 84
6040 EN=VAL(EN$):IFEN>NOREN<LOREN<>INT(EN
):THEN6030 :rem 131
6070 POKE214,8:PRINT:POKE211,1:PRINT"DATO
N."EN :rem 90
6075 POKE214,10:PRINT:POKE211,1:PRINT"VAL
ORE ="SA(EN) :rem 207
6080 POKE214,12:PRINT:POKE211,1:INPUT"NUO
VO VALORE";C:SA(EN)=C :rem 98
6090 POKE214,14:PRINT:POKE211,1:PRINT"
{ 2 GIU'}ALTRI ERRORI (S/N)?":GOSUB7
010 :rem 165
6110 IFZ$="S"THENPRINT"{CLR}":GOTO6000 :rem 117
6120 GOTO587 :rem 166
6130 GOSUB5740:POKE214,0:PRINT:POKE211,0 :rem 150
6140 IFKL=0THENPRINT"QUESTI SONO I PRIMI
{ 3 SPAZI}DIECI VALORI." :K=1:GOTO615
0 :rem 205

```

```

6145 PRINT"QUESTI SONO I DIECI{ 3 SPAZI}
VALORI SUCCESSIVI." :rem 94
6150 POKE214,3:PRINT:POKE211,1:PRINT"N. D
ATO - VALORE" :rem 183
6160 FF=0:FORK=KTOK+9:FF=FF+1 :rem 247
6165 IFK>100THENK=K+9:NEXT:GOTO587 :rem 113
6170 POKE214,FF+5:PRINT:POKE211,1:PRINTK;
TAB(11)SA(K) :rem 212
6180 NEXT :rem 14
6190 GOTO5810 :rem 215
7010 GETZ$:IFZ$="OR(Z$<>"S"ANDZ$<>"N")TH
EN7010 :rem 195
7020 RETURN :rem 169

```

La statistica diventa facile versione per C64

```

80 POKE53280,0:POKE53281,0 :rem 189
90 DIMSA(300) :rem 187
120 PRINT"{CLR}[<1>":POKE214,10:PRINT:PO
KE211,13:PRINT"{RVS}LA{ 2 SPAZI}STATI
STICA" :rem 252
140 POKE214,14:PRINT:POKE211,13:PRINT"
{RVS}DIVENTA FACILE" :rem 12
180 FORK=1TO4000:NEXT :rem 25
200 PRINT"{CLR}":POKE214,4:PRINT:POKE211,
7:PRINT"{CYN}QUESTO PROGRAMMA CALCOLA
I" :rem 230
210 POKE214,5:PRINT:POKE211,8:PRINT"SEGUE
NTI VALORI PARTENDO" :rem 123
220 POKE214,6:PRINT:POKE211,10:PRINT"DAI
DATI INTRODOTTI:" :rem 82
240 POKE214,10:PRINT:POKE211,13:PRINT"1.
VALOR MEDIO" :rem 153
250 POKE214,12:PRINT:POKE211,13:PRINT"2.
DEVIATIONE" :rem 153
260 POKE214,13:PRINT:POKE211,16:PRINT"STA
NDARD" :rem 161
290 POKE214,15:PRINT:POKE211,13:PRINT"3.
MEDIANA" :rem 162
300 POKE214,17:PRINT:POKE211,13:PRINT"4.
VARIAZIONE" :rem 166
310 POKE214,22:PRINT:POKE211,9:PRINT"PREM
I {RVS}C{OFF} PER CONTINUARE" :rem 97
320 GETZ$:IFZ$<>"C"THEN320 :rem 255
325 SUM=0:MEAN=0:DFF=0:SDDEV=0:RG=0 :rem 152
330 PRINT"{CLR}":POKE214,12:PRINT:POKE211
,4:PRINT"HAI BISOGNO DI ISTRUZIONI (S
/N)?" :rem 168
340 GOSUB7010 :rem 223
350 IFZ$="S"THENGOSUB5010 :rem 238
410 PRINT"{CLR}":POKE214,10:PRINT:POKE211
,11:INPUT"QUANTI VALORI";N :rem 14
420 IFN>300ORN<=1THENFORI=1488TO1498:POKE
I,32:NEXT:GOTO410 :rem 139
425 PRINT"{CLR}" :rem 255
430 POKE214,2:PRINT:POKE211,1:PRINT"SE CO
MMETTI UN ERRORE, CONTINUA A" :rem 33
440 POKE214,4:PRINT:POKE211,1:PRINT"IMPOS
TARE I DATI. POTRAI FARE LE" :rem 181
450 POKE214,6:PRINT:POKE211,1:PRINT"CORRE
ZIONI PIU' TARDI." :rem 168
500 FORI=1TON :rem 36
520 POKE214,11:PRINT:POKE211,10:PRINT"DAT
O N.{ 4 SPAZI}{ 4 SIN}"I:INPUTR$ :rem 35
530 SA(I)=VAL(R$) :rem 173
550 FORJ=1526TO1543:POKEJ,32:NEXT:NEXT :rem 124

```

LOAD

La statistica
diventa facile



```

560 PRINT "{CLR}":POKE214,12:PRINT:POKE211
    ,3 :rem 177
561 PRINT"VUOI FARE DELLE CORREZIONI (S/N
    )?" :rem 152
570 GOSUB7010 :rem 228
580 IFZ$="S"THENPRINT "{CLR}":GOTO6000
    :rem 74
587 PRINT "{CLR}":POKE214,11:PRINT:POKE211
    ,13:PRINT "{RVS}ATTENDI PREGO"
    :rem 141
588 POKE214,13:PRINT:POKE211,7:PRINT"CALC
    OLO DELLE STATISTICHE" :rem 213
590 FORI=1TON :rem 45
600 SUM=SUM+SA(I) :rem 22
610 NEXT :rem 214
620 MEAN=SUM/N :rem 104
630 FORI=1TON :rem 40
640 DFF=DFF+(SA(I)-MEAN)E2 :rem 255
650 NEXT :rem 218
660 SDDEV=SQR(DFF/(N-1)) :rem 146
670 FL=0 :rem 187
680 FORI=1TON-1 :rem 139
690 IFSA(I)<=SA(I+1)THEN740 :rem 41
700 Q=SA(I) :rem 83
710 SA(I)=SA(I+1) :rem 141
720 SA(I+1)=Q :rem 177
730 FL=1 :rem 154
740 NEXTI :rem 35
750 IFFL=1THEN670 :rem 247
760 RG=SA(N)-SA(1) :rem 233
765 LR=SA(1) :rem 147
767 HR=SA(N) :rem 174
810 IFN/2<>INT(N/2)THEN814 :rem 8
811 IFSA(N/2)=SA(N/2+1)THENMDD=SA(N/2)
    :rem 191
812 IFSA(N/2)<>SA(N/2+1)THENMDD=(SA(N/2)+
    SA(N/2+1))/2 :rem 202
813 GOTO1310 :rem 154
814 MDD=SA(INT(N/2+1)) :rem 219
1310 PRINT "{CLR}":POKE214,3:PRINT:POKE211
    ,9:PRINT "RISULTATI DEI CALCOLI"
    :rem 70
1320 POKE214,4:PRINT:POKE211,7:PRINT"****
    *****" :rem 56
1330 POKE214,6:PRINT:POKE211,7:PRINT"NUME
    RO DATI"SPC(10)N :rem 255
1340 POKE214,8:PRINT:POKE211,7 :rem 25
1345 PRINT "MEDIA"SPC(16)INT(MEAN*10000+.5
    )/10000 :rem 194
1350 POKE214,10:PRINT:POKE211,7 :rem 67
1355 PRINT "DEV. STANDARD"SPC(8)INT(SDDEV*
    10000+.5)/10000 :rem 231
1360 POKE214,12:PRINT:POKE211,7 :rem 70
1365 PRINT "MEDIANA"SPC(14)INT(MDD*10000+.
    5)/10000 :rem 5
1370 POKE214,14:PRINT:POKE211,7 :rem 73
1372 PRINT "VARIANZA"SPC(11)INT(RG*10000
    +.5)/10000 :rem 205
1375 POKE214,16:PRINT:POKE211,7:PRINT"VAL
    ORE PIU' BASSO"SPC(4)LR :rem 186
1377 POKE214,18:PRINT:POKE211,7:PRINT"VAL
    ORE PIU' ALTO"SPC(5)HR :rem 115

```

```

1480 POKE214,22:PRINT:POKE211,13:PRINT"
    {RVS}{YEL}PREMI UN TASTO{CYN}"
    :rem 124
1490 GETA$:IFA$=""THEN1490 :rem 191
1510 PRINT "{CLR}":POKE214,12:PRINT:POKE211
    ,4 :rem 222
1520 PRINT"VUOI ELABORARE ALTRI DATI (S/N
    )?" :rem 103
1530 GOSUB7010 :rem 17
1535 IFZ$="N"THENPRINT "{CLR}":END
    :rem 78
1540 FORI=1TON:SA(I)=0:NEXT:GOTO325
    :rem 180
5010 PRINT "{CLR}":POKE214,3:PRINT:POKE211
    ,4 :rem 173
5020 PRINT"IL MASSIMO NUMERO DI DATI CHE
    PUOI" :rem 216
5030 POKE214,5:PRINT:POKE211,1:PRINT"IMPO
    STARE E' 300, IL MINIMO E' 2."
    :rem 36
5050 POKE214,9:PRINT:POKE211,4:PRINT"IL V
    ALOR MEDIO E' LA MEDIA" :rem 4
5070 POKE214,11:PRINT:POKE211,1:PRINT"ARI
    TMETICA DEI NUMERI IMPOSTATI."
    :rem 185
5100 POKE214,15:PRINT:POKE211,4:PRINT"LA
    DEVIATIONE STANDARD E' UNA" :rem 105
5120 POKE214,17:PRINT:POKE211,1:PRINT"MIS
    URA DI QUANTO I NUMERI SI" :rem 57
5130 POKE214,19:PRINT:POKE211,1:PRINT"DIS
    COSTANO DALLA MEDIA." :rem 52
5140 PRINTTAB(5)"{ 2 GIU' }{RVS}{GRN}PREMI
    UN TASTO PER CONTINUARE{CYN}"
    :rem 115
5150 GETA$:IFA$=""THEN5150 :rem 185
5155 PRINT "{CLR}" :rem 52
5160 POKE214,1:PRINT:POKE211,4:PRINT"I DA
    TI TENDONO A CREARE UNA CURVA"
    :rem 248
5180 POKE214,3:PRINT:POKE211,1:PRINT"A FO
    RMA DI CAMPANA (DISTRIBUZIONE"
    :rem 102
5200 POKE214,5:PRINT:POKE211,1:PRINT"NO
    RMALE). LA DEVIATIONE STANDARD E' LA"
    :rem 68
5220 POKE214,7:PRINT:POKE211,1:PRINT"MISU
    RA DELL'AREA SOTTO LA CURVA."
    :rem 37
5230 POKE214,9:PRINT:POKE211,5:PRINT"N. D
    I DEV. ST."SPC(6)"% AREA" :rem 255
5245 POKE214,10:PRINT:POKE211,5:PRINT"---
    -----"SPC(6)"-----" :rem 136
5250 PRINTTAB(10)"{GIU'}1"SPC(15)"68.3"
    :rem 24
5260 PRINTTAB(10)"{GIU'}2"SPC(15)"95.5"
    :rem 28
5270 PRINTTAB(10)"{GIU'}3"SPC(15)"99.7"
    :rem 36
5280 PRINTTAB(10)"{GIU'}4"SPC(15)"99.9"
    :rem 40
5290 PRINTTAB(5)"{ 2 GIU' }{RVS}{GRN}PREMI
    UN TASTO PER CONTINUARE{CYN}"
    :rem 121
5300 GETA$:IFA$=""THEN5300 :rem 179
5430 PRINT "{CLR}" :rem 48
5435 POKE214,3:PRINT:POKE211,4:PRINT"LA M
    EDIANA E' IL VALORE CHE" :rem 59
5450 POKE214,5:PRINT:POKE211,1:PRINT"DIVI
    DE A META' I DATI." :rem 254
5490 POKE214,9:PRINT:POKE211,4:PRINT"LA V
    ARIANZA E' LA DIFFERENZA TRA"
    :rem 111

```

DAI MEMORIA AL TUO VIC-20



VIXEN SWITCHABLE 16K RAM



CON QUESTA CARTRIDGE POTRAI ESPANDERE LA MEMORIA DEL TUO VIC-20 COME VORRAI: 3K-8K-16K

Progettata e prodotta in Inghilterra dalla Stonechip Electronics questa cartuccia, tramite il semplice posizionamento di micro-switch, ti consente un ampliamento differenziato della RAM: 3K o 8K o 16K, secondo quanto richiede il programma che devi far girare. I contatti placcati in oro, assicurano una lunga durata a questa cartuccia che darà nuova vita al tuo VIC-20. Un prezzo introvabile e garanzia per 6 mesi.

LIGHT-PEN PER VIC-20 E CBM-64 A L. 54.000 + IVA
della Stonechip Electronics; ottima stabilità e risoluzione verticale/orizzontale. Garanzia di 6 mesi.

CARTRIDGE COMMODORE A L. 24.500 + IVA
CBM-64: 601 Jupiter lander - 603 Sea Wolf - 606 Clowns. **VIC-20:** 01 Avenger - 07 Jupiter lander - 10 Rat race - 12 Mole attack - 13 Ride on Fort Knox - 18 Voodoo Castle - 19 Sar-gon chess II - 22 Cosmic cruncher - 24 Omega race.

MODULO DI ESPANSIONE PER VIC-20 A LIRE 62.000 + IVA
Questo modulo di espansione differenziata, prodotto dalla Stonechip Electronics, ti consente di utilizzare contemporaneamente (posizionando i microswitch di cui è dotato) da 2 a 4 cartridges. Aggiunge inoltre alla ROM 4K. Inserendo con l'espansione Vixen 16K Ram una espansione standard di 8K si ottengono ben 24K Ram. E' anche dotato di una presa per collegare un ulteriore modulo di espansione. Garanzia 6 mesi.

UN OMAGGIO A CHI ORDINA ALMENO DUE ARTICOLI

Spedire il presente **MODULO D'ORDINE**, o fotocopia, in una busta, unendo L. 2.600 in francobolli per spese postali.



Spett.le APCO s.r.l. - Cas. Post. 239 - 10015 IVREA (Torino)
desidero ricevere quanto da me indicato. Pagherò direttamente al postino gli importi qui elencati, che sono comprensivi di IVA e di spese di imballo e contrassegno.

N. Modulo di espansione per VIC-20 a Lire 75.000 caduno

N. Vixen Switchable 16K RAM a Lire 95.000 caduna

N. Light-Pen a Lire 65.000 caduna

Le seguenti Cartridge Commodore a Lire 29.500 caduna

(indicare i codici) _____

☐ Non acquisto nulla ma desidero ricevere i Vostri cataloghi che usciranno nel 1985.

Nome e Cognome _____

Via _____

Città _____ CAP. _____

data _____ firma _____

```

5510 POKE214,11:PRINT:POKE211,1:PRINT"IL
      VALORE PIU' BASSO E QUELLO PIU'"
      :rem 98
5520 POKE214,13:PRINT:POKE211,1:PRINT"ALT
      O. ESSA DA' UNA STIMA APPROSSIMATIVA
      "
      :rem 25
5530 POKE214,15:PRINT:POKE211,1:PRINT"DEL
      LO 'SPARPAGLIAMENTO' DEI DATI."
      :rem 152
5540 POKE214,17:PRINT:POKE211,1:PRINT"LA
      DEV. STANDARD E' PIU' ATTENDIBILE."
      :rem 19
5580 PRINTTAB(9)"{ 2 GIU' }{RVS}{GRN}PREMI
      {OFF}I{RVS} PER INIZIARE{CYN}"
      :rem 161
5600 GETAS:IFAS="OR(AS<>"I")THEN5600
      :rem 23
5610 RETURN
      :rem 172
5810 GETZ$:IFZ$="OR(Z$<>"C"ANDZ$<>"P"AND
      Z$<>"F")THEN5810
      :rem 22
5820 FL=0:IFZ$="P"THENFL=1:GOTO6130
      :rem 82
5830 IFZ$="F"THEN587
      :rem 119
5840 PRINT"{CLR}"
      :rem 53

6000 POKE214,12:PRINT:POKE211,1:PRINT"RIC
      ORDI IL N. DEL DATO SBAGLIATO (S/N)?
      "
      :rem 87
6010 GOSUB7010
      :rem 15
6020 IFZ$="N"THEN6130
      :rem 157
6030 PRINT"{CLR}":POKE214,6:PRINT:POKE211
      ,10:INPUT"NUMERO DEL DATO";ENS
      :rem 179
6040 EN=VAL(ENS):IFEN>NOREN<LOREN<>INT(EN)
      )THEN6030
      :rem 131
6070 POKE214,8:PRINT:POKE211,6:PRINT"DATO
      N."ENSPC(5)"VALORE ="SA(EN)
      :rem 141
6080 POKE214,11:PRINT:POKE211,6:INPUT"NUO
      VO VALORE";C
      :rem 52
6090 SA(EN)=C
      :rem 199
6096 POKE214,14:PRINT:POKE211,6:PRINT"ALT
      RE CORREZIONI (S/N)?"
      :rem 57
6100 GOSUB7010
      :rem 15
6110 IFZ$="S"THENPRINT"{CLR}":GOTO6000
      :rem 117
6120 GOTO587
      :rem 166

6130 PRINT"{CLR}"
      :rem 46
6132 POKE214,21:PRINT:POKE211,0
      :rem 65
6134 PRINT"{RVS}{WHT}{ 2 SPAZI}C=CORREGGI
      { 5 SPAZI}P=PROSEGUI{ 5 SPAZI}F=FINE
      { 2 SPAZI}";
      :rem 99
6137 POKE214,2:PRINT:POKE211,2
      :rem 23

6140 IFFL=0THENPRINT"QUESTI SONO I PRIMI
      DIECI VALORI.";K=1:GOTO6150:rem 200
6145 PRINT"QUESTI SONO I DIECI VALORI SUC
      CESSIVI."
      :rem 94
6150 POKE214,6:PRINT:POKE211,10:PRINT"DAT
      O"SPC(12)"VALORE"
      :rem 31
6160 FF=0:FORK=KTOK+9:FF=FF+1
      :rem 247

6165 IFK>300THENK=K+9:NEXT:GOTO587
      :rem 115
6170 POKE214,FF+7:PRINT:POKE211,10:PRINTK
      ;TAB(27)SA(K)
      :rem 13
6180 NEXT
      :rem 14
6190 GOTO5810
      :rem 215
7010 GETZ$:IFZ$="OR(Z$<>"S"ANDZ$<>"N")TH
      EN7010
      :rem 195
7020 RETURN
      :rem 169

```



Lightsaver

di J. Rhees

trad. e adatt.

di S. Albarelli e M. Anticoli

Le lampadine molto spesso creano seri problemi e bisogna di tanto in tanto sostituirle. Inoltre, quando ci sfuggono dalle mani, o si staccano dal lampadario, cadendo per terra si frantumano in mille pezzi.

Questo gioco di prontezza e di attenzione utilizza le paddle e il linguaggio macchina per velocizzare l'azione e per sfruttare al meglio le possibilità grafiche e sonore di quel fantastico computer qual è il C64 e del suo "fratello minore" VIC 20.

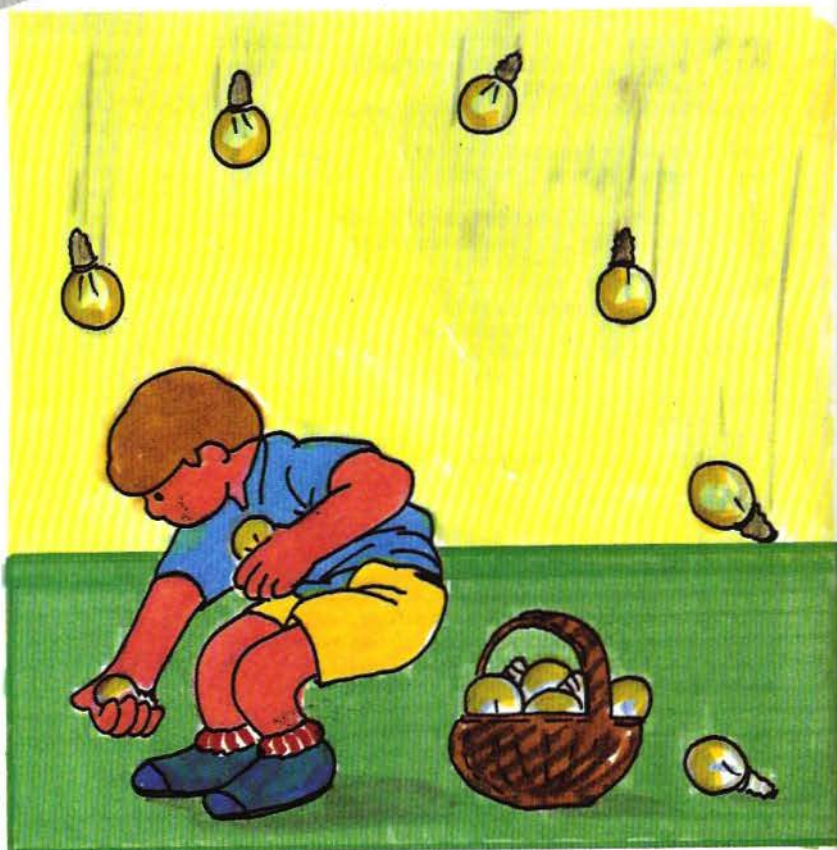
Il povero Joe è alle prese con il suo capriccioso lampadario, che non ne vuole sapere di trattenere le lampadine che il nostro amico gli applica ripetutamente. Perciò la moglie di Joe, preoccupata di tener pulita la casa, tenta di non far cadere le lampadine per terra, cercando disperatamente di farle cadere in tre cesti che ha trovato rimescolando nel suo sconfusionato sgabuzzino. Però la moglie del povero Joe non è molto giovane e i suoi riflessi non sono più quelli di una volta.

Quando le cade per terra una lampadina deve raccogliere i frammenti di vetro con uno dei cestini a sua disposizione, che poi le viene tolto per qualche tempo dal figliolo giudizioso che butta i frammenti nella spazzatura.

Perché non aiutarla nella sua faticosa impresa di pulizia domestica??? Con le paddle inserite nella porta 1 è possibile muovere i cesti, che possono essere di due larghezze differenti, selezionabili sia all'inizio della partita che durante il gioco.

La partita è organizzata in livelli ed è possibile selezionare, all'inizio, uno dei sedici gradi di difficoltà disponibili (richiesti mediante una lettera da A a P dell'alfabeto inglese).

Ad ogni livello aumenta sempre più la velocità di caduta, il numero di lampadine da raccogliere e il punteggio ottenuto per ogni lampadina salvata. Ogni 2000 punti si riceve un cestino "BONUS" (se non se ne possiedono già più di tre).



Ogni volta che una lampadina cadrà a terra si perderà un cestino e si tornerà indietro di un livello.

Alla fine della partita, se si è realizzato il maggior punteggio ottenuto, è possibile memorizzare le proprie iniziali utilizzando le paddle.

La versione per C64 utilizza in pieno tutte le possibilità del computer: sprite, colori, ridefinizione dei caratteri, musica ad alto livello. Essa impiega due routine in l.m. interattivamente con il BASIC: la prima per gestire le animazioni del gioco e la seconda per trasferire velocemente i caratteri della ROM alla RAM, per poi programmarli.

Inoltre nel programma è inclusa una dimostrazione di gioco, nella quale il com-

puter si cimenterà.

Le prime due voci del SID (Sound Interface Device) sono usate per produrre gli effetti sonori del gioco, mentre la terza fornisce valori casuali alle subroutine in linguaggio macchina.

Note alla versione per VIC 20

La versione per il VIC 20 INESPANSO è scritta interamente in linguaggio macchina e offre cinque livelli di gioco.

Anche per la versione VIC 20 si utilizzano le paddle per muovere i cestini. Per trascriverla è necessario utilizzare il programma MLX pubblicato in altra parte della rivista.

Se non si è in possesso della cassetta

allegata alla rivista, è necessario, prima di caricare l'MLX, battere le seguenti istruzioni BASIC in forma diretta, ossia senza numero di linea:

POKE 44, 35: POKE 8960, 0: NEW

poi, dopo aver caricato in memoria il programma MLX e averlo fatto girare, bisogna rispondere nel modo seguente

alle domande proposte dal computer:

Inizio = 6430

Fine = 7678

Dopo aver salvato il programma in linguaggio macchina è possibile caricarlo con un semplice:

LOAD "nome prog.", 1, 1 per il nastro e LOAD "nome prog.", 8, 1 per il disco. Per farlo girare basta dare un semplice RUN.

Note alla versione per C64

Nel listato della versione di "Lightsaver" per C64 compare a volte il simbolo "~". Questo simbolo, per una limitazione della nostra stampante, corrisponde al tasto "£" (simbolo della Lira). Quando incontrate "~" premete dunque il tasto "£".

MLX ridotto per VIC 20

```

210 S=6430:E=7678           :rem 136
300 PRINT "{CLR}";CHR$(14):AD=S :rem 56
310 PRINTRIGHT$( "0000"+MID$(STR$(AD),2),5 :rem 234
   );":":FORJ=1TO6
320 GOSUB570:IFN=-1THENJ=J+N:GOTO320 :rem 228
   :rem 168
480 IFN<0THENPRINT:GOTO310 :rem 168
490 A(J)=N:NEXTJ :rem 199
500 CKSUM=AD-INT(AD/256)*256:FORI=1TO6:CK :rem 200
   SUM=(CKSUM+A(I))AND255:NEXT :rem 200
510 PRINTCHR$(18);:GOSUB570:PRINTCHR$(20) :rem 234
   :rem 255
515 IFN=CKSUMTHEN530 :rem 255
520 PRINT:PRINT"ERRORE!":PRINT:GOSUB1000: :rem 104
   GOTO310 :rem 104
530 GOSUB2000 :rem 218
540 FORI=1TO6:POKEAD+I-1,A(I):NEXT :rem 80
   :rem 212
550 AD=AD+6:IFAD<ETHEN310 :rem 109
560 GOTO720 :rem 88
570 N=0:Z=0 :rem 79
580 PRINT "[<+>]"; :rem 95
581 GETA$:IFA$=""THEN581 :rem 229
585 PRINTCHR$(20);:A=ASC(A$):IFA=13ORA=44 :rem 137
   ORA=32THEN670 :rem 10
590 IFA>128THENN=-A:RETURN :rem 172
600 IFA>20THEN630 :rem 109
610 GOSUB690:IFI=1ANDT=44THENN=-1:PRINT" :rem 105
   { 2 SIN}";:GOTO690 :rem 106
620 GOTO570 :rem 229
630 IFA<48ORA>57THEN580 :rem 71
640 PRINTA$;:N=N*10+A-48 :rem 114
650 IFN>255THENA=20:GOSUB1000:GOTO600 :rem 240
   :rem 149
660 Z=Z+1:IFZ<3THEN580 :rem 68
670 IFZ=0THENGOSUB1000:GOTO570 :rem 205
680 PRINT",":RETURN :rem 7
690 S%=PEEK(209)+256*PEEK(210)+PEEK(211) :rem 133
   :rem 149
692 FORI=1TO3:T=PEEK(S%-I) :rem 68
695 IFT<>44ANDT<>58THENPOKES%-I,32:NEXT :rem 205
   :rem 91
700 PRINTLEFT$("{ 3 SIN}",I-1);:RETURN :rem 158
   :rem 3
720 INPUT "{CLR} {GIU'} NOME FILE";F$ :rem 115
   :rem 91
730 PRINT:PRINT "{ 2 GIU'} {RVS} N{OFF}ASTRO :rem 158
   O {RVS}D{OFF}ISCO?" :rem 115
740 GETA$:IFA$=<>"N"ANDA$=<>"D"THEN740 :rem 91
   :rem 158
750 DV=1-7*(A$="D"):IFDV=8THENF$="0":F$ :rem 3
   :rem 109
760 TS=F$:ZK=PEEK(53)+256*PEEK(54)-LEN(T$) :rem 109
   ):POKE782,ZK/256 :rem 3
762 POKE781,ZK-PEEK(782)*256:POKE780,LEN(T$) :rem 109
   ):SYS65496 :rem 109
763 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 69
   66

```

```

765 POKE254,S/256:POKE253,S-PEEK(254)*256 :rem 12
   :POKE780,253 :rem 12
766 POKE782,E/256:POKE781,E-PEEK(782)*256 :rem 124
   :SYS65496 :rem 111
770 IF(PEEK(783)AND1)OR(ST AND191)THEN780 :rem 122
   :rem 193
775 END :rem 122
780 PRINT "{GIU'} ERRORE! RIPROVA!":GOTO720 :rem 166
   :rem 166
845 POKE780,1:POKE781,DV:POKE782,1:SYS654 :rem 74
   66 :rem 70
1000 REM :rem 165
1001 POKE36878,15:POKE36874,190 :rem 206
1002 FORW=1TO300:NEXTW :rem 117
1003 POKE36878,0:POKE36874,0:RETURN :rem 119
2000 REM :rem 166
2001 FORW=15TO0STEP-1:POKE36878,W:POKE368 :rem 22
   76,240:NEXTW :rem 119
2002 POKE36876,0:RETURN :rem 119

```

Lightsaver versione per VIC 20

```

6430 :011,016,000,000,158,052,011
6436 :049,048,057,000,000,000,190
6442 :169,000,141,092,003,141,076
6448 :093,003,169,024,141,015,237
6454 :144,032,078,020,032,100,204
6460 :016,032,236,017,032,042,179
6466 :017,169,007,141,074,003,221
6472 :169,003,141,064,003,173,113
6478 :076,003,016,024,032,203,176
6484 :017,172,074,003,196,251,029
6490 :208,003,076,077,016,032,246
6496 :015,019,032,246,018,169,083
6502 :010,141,076,003,206,076,102
6508 :003,032,108,017,032,125,169
6514 :016,032,038,019,032,179,174
6520 :019,173,072,003,208,207,034
6526 :076,205,018,169,000,141,223
6532 :060,003,141,061,003,141,029
6538 :074,003,032,042,017,169,219
6544 :003,141,072,003,169,010,030
6550 :141,076,003,096,169,000,123
6556 :141,019,145,169,127,141,130
6562 :034,145,173,008,144,074,228
6568 :074,074,074,073,015,056,022
6574 :233,003,016,002,169,000,085
6580 :141,064,003,168,162,002,208
6586 :185,162,031,208,008,169,181
6592 :032,153,162,031,032,023,113
6598 :017,185,184,031,208,008,063

```

LOAD



Lightsaver



6604 :169,032,153,184,031,032,037
 6610 :023,017,185,206,031,208,112
 6616 :008,169,032,153,206,031,047
 6622 :032,023,017,200,202,016,200
 6628 :213,172,064,003,196,252,104
 6634 :240,071,164,252,169,032,138
 6640 :162,003,153,184,031,153,158
 6646 :206,031,200,202,016,246,123
 6652 :172,064,003,169,027,153,072
 6658 :184,031,153,206,031,132,227
 6664 :252,169,000,153,184,151,149
 6670 :153,206,151,200,169,028,153
 6676 :153,184,031,153,206,031,010
 6682 :169,000,153,184,151,153,068
 6688 :206,151,200,169,029,153,172
 6694 :184,031,153,206,031,169,044
 6700 :000,153,184,151,153,206,123
 6706 :151,096,173,082,003,240,027
 6712 :003,206,082,003,238,060,136
 6718 :003,173,060,003,208,003,000
 6724 :238,061,003,162,004,160,184
 6730 :016,024,032,240,255,174,047
 6736 :060,003,173,061,003,032,156
 6742 :205,221,056,173,060,003,036
 6748 :237,092,003,133,170,173,132
 6754 :061,003,237,093,003,005,244
 6760 :170,144,012,173,060,003,154
 6766 :141,092,003,173,061,003,071
 6772 :141,093,003,162,008,160,171
 6778 :016,024,032,240,255,174,095
 6784 :092,003,173,093,003,032,012
 6790 :205,221,096,169,206,133,140

6796 :167,169,031,133,168,173,213
 6802 :141,002,208,251,169,228,121
 6808 :133,170,169,031,133,171,191
 6814 :169,228,133,180,169,151,164
 6820 :133,181,162,021,160,021,074
 6826 :177,167,208,012,072,169,207
 6832 :032,145,167,104,145,170,171
 6838 :169,000,145,180,136,016,060
 6844 :237,056,165,167,233,022,044
 6850 :133,167,165,168,233,000,036
 6856 :133,168,056,165,170,233,101
 6862 :022,133,170,165,171,233,076
 6868 :000,133,171,056,165,180,149
 6874 :233,022,133,180,165,181,108
 6880 :233,000,133,181,202,016,221
 6886 :193,096,032,148,224,165,064
 6892 :142,162,003,010,202,208,195
 6898 :252,024,101,142,010,101,104
 6904 :142,105,003,133,142,074,079
 6910 :074,056,233,012,176,252,033
 6916 :105,015,133,251,096,169,005
 6922 :128,141,019,145,169,255,099
 6928 :141,034,145,169,147,032,172
 6934 :210,255,162,011,160,005,057
 6940 :024,032,240,255,160,000,227
 6946 :185,193,019,032,210,255,160
 6952 :200,192,012,208,245,169,042
 6958 :014,032,210,255,162,013,220
 6964 :160,005,024,032,240,255,000
 6970 :160,000,185,205,019,032,147
 6976 :210,255,200,192,012,208,117
 6982 :245,032,228,255,240,251,041
 6988 :201,049,144,247,201,054,204
 6994 :176,243,056,233,048,073,143
 7000 :255,141,063,003,169,147,098
 7006 :032,210,255,174,063,003,063
 7012 :024,169,000,105,048,202,136
 7018 :208,251,141,082,003,162,185
 7024 :003,160,016,024,032,240,075
 7030 :255,160,000,185,010,020,236
 7036 :032,210,255,200,192,007,252
 7042 :208,245,162,006,160,016,159
 7048 :024,032,240,255,160,000,079
 7054 :185,017,020,032,210,255,093
 7060 :200,192,012,208,245,162,143
 7066 :010,160,016,024,032,240,124
 7072 :255,160,000,185,029,020,041
 7078 :032,210,255,200,192,007,038
 7084 :208,245,162,013,160,016,208
 7090 :024,032,240,255,160,000,121
 7096 :185,036,020,032,210,255,154
 7102 :200,192,015,208,245,162,188
 7108 :022,160,001,024,032,240,163
 7114 :255,160,000,185,059,020,113
 7120 :032,210,255,200,192,019,092
 7126 :208,245,173,017,145,041,019
 7132 :016,208,249,160,021,169,019

```

7138 :032,153,228,031,136,208,246
7144 :248,096,160,002,162,022,154
7150 :024,032,240,255,160,000,181
7156 :185,217,019,032,210,255,138
7162 :200,192,018,208,245,032,121
7168 :228,255,240,251,201,089,240

7174 :240,007,201,078,240,006,010
7180 :076,226,018,076,029,016,197
7186 :000,164,251,140,074,003,138
7192 :162,001,024,032,240,255,226
7198 :160,000,185,235,019,032,149
7204 :210,255,200,192,016,208,093
7210 :245,096,172,074,003,162,026
7216 :001,024,032,240,255,160,248
7222 :000,185,251,019,032,210,239
7228 :255,200,192,014,208,245,150
7234 :096,160,021,185,228,031,019
7240 :208,028,169,036,153,228,126
7246 :031,169,000,153,228,151,042
7252 :072,152,072,032,142,019,061
7258 :032,179,019,104,168,104,184
7264 :206,072,003,076,076,019,036
7270 :136,208,220,169,032,153,252
7276 :228,031,162,011,160,016,204
7282 :024,032,240,255,173,082,152
7288 :003,205,084,003,240,013,156
7294 :160,000,185,051,020,032,062
7300 :210,255,200,192,008,208,181
7306 :245,173,082,003,141,084,098
7312 :003,056,073,255,170,169,102
7318 :000,032,205,221,162,015,017
7324 :160,016,024,032,240,255,115
7330 :174,072,003,169,000,032,100
7336 :205,221,096,169,015,141,247
7342 :014,144,169,129,141,013,016
7348 :144,162,010,138,072,032,226
7354 :179,019,104,170,202,208,044
7360 :246,169,000,141,014,144,138
7366 :024,173,082,003,105,016,089
7372 :141,082,003,096,172,082,012
7378 :003,162,080,202,208,253,094
7384 :136,192,255,208,246,096,069
7390 :144,076,073,071,072,084,230
7396 :083,065,086,069,082,032,133
7402 :076,069,086,069,076,032,130
7408 :040,049,045,053,041,063,019
7414 :080,076,065,089,032,065,141
7420 :071,065,073,078,032,089,148
7426 :032,079,082,032,078,063,112
7432 :031,094,157,017,094,157,046
7438 :157,017,095,033,035,017,112
7444 :157,157,144,064,144,032,206
7450 :157,017,032,157,157,017,051
7456 :032,032,032,017,157,157,203
7462 :017,028,083,067,079,082,138
7468 :069,144,030,072,073,017,193
7474 :157,157,083,067,079,082,163

```

```

7480 :069,144,156,083,080,069,145
7486 :069,068,144,066,085,076,058
7492 :066,083,017,157,157,157,193
7498 :157,157,076,069,070,084,175
7504 :032,032,032,032,157,157,010
7510 :157,157,080,082,069,083,202
7516 :083,032,070,073,082,069,245
7522 :032,084,079,032,083,084,236
7528 :065,082,084,160,000,185,168
7534 :000,128,153,000,028,200,107
7540 :208,247,160,000,185,000,148
7546 :129,153,000,029,200,208,073
7552 :247,160,007,185,150,020,129
7558 :153,000,028,136,016,247,202
7564 :160,039,185,158,020,153,087
7570 :216,028,136,016,247,160,181
7576 :007,185,198,020,153,008,211
7582 :029,136,016,247,160,015,249
7588 :185,206,020,153,024,029,013
7594 :136,016,247,169,255,141,110
7600 :005,144,096,024,024,024,237
7606 :036,066,066,102,060,000,000
7612 :000,234,117,058,029,014,128
7618 :007,000,000,170,085,170,114
7624 :085,170,255,000,000,175,117
7630 :094,188,120,240,224,024,072
7636 :024,024,024,024,024,024,100
7642 :024,000,001,003,007,015,012
7648 :031,063,127,255,255,255,186
7654 :255,255,255,255,255,000,225
7660 :128,192,224,240,248,252,240
7666 :254,000,024,024,060,086,178
7672 :203,000,000,000,000,000,195

```

Lightsaver versione per C64

```

1 GOSUB550 :rem 75
2 Z=646:POKE53280,0:POKE53281,0:POKEZ,1:X
  =58692:SYSX:V=53248 :rem 89
3 G=54272:POKEG+6,0:POKEG+5,0:POKEG+4,0:P
  OKEG+6,240:POKEG+4,17:POKEG+24,143
  :rem 122
4 POKEG+6,240:NS$="000000":GOSUB400
  :rem 187
10 T=49152:H=54273:B=832:C=1022:E=49550:F
  ORA=BTOC:READM:POKEA,M:POKEH,M:NEXT
  :rem 131
20 B=V+39:FORA=TT0E:READM:POKEA,M:POKEH,M
  :POKEB,M:NEXT:POKEG+5,186:POKEG+6,.
  :rem 124
26 POKE252,208:POKE251,..:POKE253,..:BL=872
  :POKE254,48 :rem 240
28 POKE56334,..:POKE1,51:SYS49490:POKE4951
  3,56:POKE49500,145:POKE49501,253
  :rem 208
29 POKE49502,200:POKE49503,234:SYS49490:P
  OKE1,55:POKE56334,1:GOTO600 :rem 214
60 POKE53280,3:SYSX:PRINT"[ 5 GIU' ]
  [ 7 SPAZI]SCEGLI IL LIVELLO. (A-P) ";
  :rem 194
61 GETA$:SL=ASC(A$+" ") -59:IFSL<60RSL>21T
  HEN61 :rem 170

```



```

62 PRINTAS:POKE53280,14:PRINT:PRINT"
{ 3 GIU'}SCEGLI LA LARGHEZZA DEL CESTO
(P/G)";:P=. :rem 225
63 GETSS:IFSS<>"P"ANDSS<>"G"THEN63
:rem 245
64 PRINTSS"{ 5 GIU'}":POKE53280,6:LV=. :PO
KE209,88:POKE210,6:IFBW=141THENGOSUB50
0 :rem 146
65 PRINT"PREMI <F1> PER CAMBIARE DURANTE
IL GIOCO":GOSUB700 :rem 47
67 POKE63,.:POKE65,.:POKE67,.:SYSX:SS="0
00000":GOSUB202 :rem 60
68 POKEV+29,-(SS="G"):GOSUB340:GOTO71
:rem 158
70 GOSUB200 :rem 121
71 PRINT"[HOME]{ 17 GIU'}{ 2 SPAZI}{GRN}P
REMI FIRE PER GIOCAR.....!":rem 68
72 N=244-PEEK(54297):N=N*-(N=24)-24*(N<2
4):POKEV,N :rem 93
73 POKEV+29,(PEEK(V+29)-(PEEK(197)=4))AND
1:IF(PEEK(56321)AND4)THEN72 :rem 158
74 PRINT"[SU]{ 31 SPAZI}":POKE254,.:SYST:
IFPEEK(68)THEN80 :rem 252
76 M=54273:POKEF+11,.:N=5:I=125:J=53280:F
ORA=NTOSTEP3:POKEJ,A :rem 15
77 POKEM,AAND(LV+5)*2:NEXT:POKEF+5,186:PO
KEF+6,.:POKEF+12,186:POKEF+13,
:rem 54
79 POKEJ,.:GOTO70 :rem 29
80 POKEF+11,.:POKEF+4,.:POKEF+1,186:POKEF
+4,129:POKEF+5,188:POKEF+6,.:rem 30
81 POKEV+4,PEEK((PEEK(781)+4)+V):POKEV+5,
232:B=50:D=-.9:E=V+3:F=10264 :rem 175
82 FORA=BTO.STEPD:POKEE,A:N=RND(.)*21:M=N
*1.4:POKEN+BL,PEEK(N+BL)ANDRND(.)*256
:rem 23
83 POKEM+F,PEEK(M+F)ORRND(.)*256:POKE780,
M:SYS49540:NEXT:POKEV+41,0:POKE53281,0
:rem 47
84 LV=LV-2:P=P+1:POKEV+2,.:LV=LV*-(LV>-1)
:FORA=BLTOBL+20:POKEA,.:rem 197
85 NEXT:BL=BL-20:IFBW=141THENGOTO600
:rem 1
86 GOSUB200:IFP<3THENGOSUB340:GOTO71
:rem 41
88 IFVAL(SS)=VAL(HS)ANDVAL(SS)>0THENGOS
UB900 :rem 99
90 POKEV+21,.:PRINT"[HOME]{ 7 GIU'}{CYN}"
SPC(9)"** GAME OVER **":BL=872:P=.
:rem 87
96 GOSUB800:GOTO600 :rem 144
100 DATA255,255,255,109,182,218,27
:rem 192
101 DATA109,180,13,182,216,11,109,176
:rem 69
102 DATA7,255,224,0,0,0,255,255,255
:rem 220
103 DATA91,109,182,45,182,216,27,109
:rem 33
104 DATA176,13,182,208,7,255,224,0,0,0
:rem 109
105 DATA255,255,255,109,182,218,27,109
:rem 139

```

```

106 DATA180,13,182,216,11,109,176,7
:rem 231
107 DATA255,224,0,0,0,0,60,0,0,60,0
:rem 22
108 DATA0,60,0,0,60,0,0,60,0,0,60,0,0
:rem 11
109 DATA60,0,0,60,0,0,60,0,0,60,0,0,60
:rem 66
110 DATA0,0,60,0,0,60,0,0,60,0,0,60,0
:rem 4
111 DATA0,20,0,0,85,0,1,85,64,5,85,80
:rem 50
112 DATA21,85,84,170,170,170,0,0,0,0,0
:rem 90
113 DATA0,0,0,0,0,0,0,0,24,0,0,60,0
:rem 247
114 DATA0,60,0,0,36,0,0,126,0,0,126,0
:rem 21
115 DATA0,255,0,1,255,128,1,255,128,3
:rem 60
116 DATA253,192,1,253,128,1,251,128,0
:rem 66
117 DATA255,0,0,60,0,0,0,0,0,0,0,0,0
:rem 49
118 DATA120,173,2,220,72,169,192,141,2
:rem 117
119 DATA220,169,64,141,0,220,160,128
:rem 19
120 DATA234,136,16,252,56,169,220,237
:rem 79
121 DATA25,212,176,2,169,0,24,105
:rem 122
122 DATA24,141,0,208,104,141,2,220,88
:rem 54
123 DATA172,30,208,169,8,133,253,162,2
:rem 123
124 DATA254,5,208,189,4,208,240,33,169
:rem 135
125 DATA188,56,253,5,208,176,13,152,37
:rem 139
126 DATA253,240,8,169,0,157,4,208,76
:rem 35
127 DATA14,193,189,5,208,201,232,208,5
:rem 128
128 DATA169,1,133,68,96,232,232,6,253
:rem 88
129 DATA224,12,240,3,76,51,192,234,164
:rem 125
130 DATA197,192,4,208,17,165,255,201
:rem 32
131 DATA64,208,11,169,1,24,109,29,208
:rem 75
132 DATA41,1,141,29,208,132,255,164,64
:rem 120
133 DATA240,99,132,64,173,2,208,24,101
:rem 120
134 DATA254,141,2,208,198,2,16,84,169
:rem 86
135 DATA31,160,0,133,2,173,27,212,133
:rem 55
136 DATA254,56,237,2,208,176,8,173,2
:rem 37
137 DATA208,56,229,254,160,1,74,74,74
:rem 91
138 DATA74,74,133,254,56,233,4,176,225
:rem 140
139 DATA192,1,208,9,165,254,169,0,56
:rem 41
140 DATA229,254,133,254,230,66,230,66
:rem 79
141 DATA166,66,224,10,208,4,162,0,134
:rem 65

```

```

142 DATA66,169,80,157,7,208,173,2,208 :rem 90
143 DATA157,6,208,234,234,206,64,0 :rem 183
144 DATA169,146,133,150,198,150,165 :rem 247
145 DATA150,141,1,212,230,9,165,9,141 :rem 63
146 DATA8,212,201,30,208,3,141,11,212 :rem 50
147 DATA172,52,3,32,204,255,32,204,255 :rem 120
148 DATA136,192,0,208,248,76,0,192,165 :rem 138
149 DATA251,248,24,101,67,133,67,169,0 :rem 135
150 DATA101,65,133,65,169,0,101,63,133 :rem 113
151 DATA63,216,162,4,181,63,72,41,15 :rem 23
152 DATA24,105,48,157,19,5,104,74,74 :rem 33
153 DATA74,74,24,105,48,157,18,5,202 :rem 32
154 DATA202,16,230,169,129,141,11,212 :rem 65
155 DATA169,0,133,9,198,252,208,3,133 :rem 84
156 DATA68,96,76,92,192,160 :rem 122
157 DATA0,177,251,17,253,145,253,177,251, :rem 86
    74,200,145,253,208,242
158 DATA230,252,230,254,165,254,201,52 :rem 123
159 DATA208,230,96,234,238,0,201,174,0 :rem 124
160 DATA201,189,0,200,168,138,105,109 :rem 71
161 DATA170,173,2,208,157,0,200,234,140,1 :rem 22
    41,41,208,141,32,208,141,33,208,96,32
162 DATA48,126,103,103,103,103,103,127 :rem 106
163 DATA63,49,28,60,124,28,28,28,127 :rem 41
164 DATA63,50,126,103,7,127,112,115,127 :rem 166
165 DATA63,51,126,103,7,63,7,103 :rem 81
166 DATA127,63,52,6,15,27,51,127,127 :rem 32
167 DATA7,15,53,126,96,126,103,7,103 :rem 29
168 DATA127,63,54,126,103,96,126,103 :rem 34
169 DATA103,127,63,55,126,103,7,14,28 :rem 78
170 DATA28,28,28,56,126,103,103,127 :rem 234
171 DATA103,103,127,63,57,126,103,103 :rem 63
172 DATA127,7,103,127,63,104,0,0,0,0 :rem 253
173 DATA239,239,85,254 :rem 125
200 NSS="" :FORA=1298TO1303:NSS=NSS+CHR$(P :rem 99
    EEK(A)):NEXT
201 AA=INT(VAL(NSS)/2000)-INT(Q/2000):IFA :rem 176
    ATHENGOSUB300
202 Q=VAL(NSS):IFQ>=VAL(HSS) THENHSS=NSS :rem 147
204 POKE53280,,:LV=LV+1:LV$=STR$(LV)+" :rem 151
    { 2 SPAZI}":IFLV>1 THEN208

```

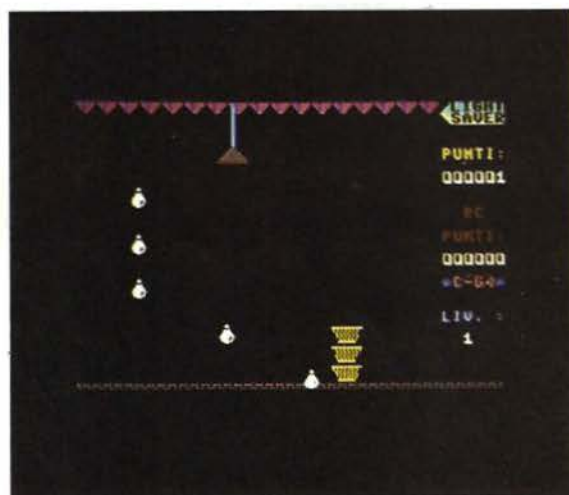
```

206 FORA=1024TO1057STEP2:POKEA,95:POKEA+1 :rem 199
    ,105:POKE54272+A,4:POKE54273+A,10:NEX
    T
207 FORA=1984TO2023:POKEA,104:POKEA+54272 :rem 138
    ,12:NEXT
208 PRINTCHR$(19)CHR$(159)CHR$(18)SPC(34) :rem 86
    CHR$(169)"LIGHT"CHR$(146):POKEZ,13
209 PRINTSPC(34)CHR$(127)CHR$(18)"SAVER": :rem 172
    POKEZ,7:PRINTSPC(74)"PUNTI:"SPC(74);
210 PRINTCHR$(5)NSS:POKEZ,8:PRINTSPC(76)" :rem 113
    RC"SPC(76)"PUNTI:"SPC(74)CHR$(5)HSS
211 PRINTSPC(34):POKEZ,6:PRINT""":POKEZ :rem 252
    ,10:PRINTHUS;POKEZ,6:PRINT""
212 POKEZ,14:PRINTSPC(74)"LIV. :SPC(75)C :rem 99
    HR$(5)LV$:F=54272:POKEF+20,128
216 POKE2040,13:POKE2041,14:POKEF+19, :rem 125
220 POKEV+1,205:POKEV+3,50:POKEV+23,3:POK :rem 156
    EV+28,2:POKEV+39,7:POKEF+18,
222 FORA=4TO14STEP2:POKEV+A,,:POKEV+A+1,A :rem 71
    *8+140:POKE2040+A/2,15:NEXT
223 FORA=10240TO10303:POKEA,,:NEXT:POKE20 :rem 34
    42,160:POKE66,,:POKE2,
226 POKEV+37,12:POKEV+38,14:FORA=41TO46:P :rem 238
    OKEV+A,1:NEXT:POKEV+21,255:POKEF+19,
228 POKEF+11,0:POKEF+7,0:POKEF+13,128:POK :rem 248
    EF+12,,:K=260-SL*LV:POKEF+20,128
230 POKE64,LV*2+9:POKE252,LV*2+9:LN=LV:PO :rem 208
    KEF+15,255:IFLN>100THENLN=100
232 POKE251,INT(LN/10)*16+LN-INT(LN/10)*1 :rem 17
    0:POKEF+14,,:POKEF+18,129:IFK<1 THENK=
    1
233 POKEF+1,,:POKEF+4,,:POKEF+5,,:POKEF+6 :rem 24
    ,128:POKEF,,:POKEF+24,143:POKEF+4,33
234 POKE820,K:POKEV+41,,:RETURN :rem 156
300 F=54272:POKEF+4,,:POKEF+5,128:POKEF+6 :rem 209
    ,128:POKEF+4,17
301 FORA=1TO4:ST(A)=PEEK(62+A):NEXT:BL=BL :rem 17
    +20*AA:IFBL>872THENBL=872
303 P=P-AA:P=P*(P>.):RESTORE :rem 251
310 RESTORE:FORA=832TO832+20*(3-P):READM :rem 18
    POKEA+1,,:POKEA,M
330 POKEF+1,INT(RND(.)*256):NEXT:FORA=1TO :rem 179
    4:POKE62+A,ST(A):NEXT:POKEF+1,,:RETUR
    N
340 POKEF+4,,:POKEF+5,128:POKEF+6,128:POK :rem 216
    EF+4,33:B=28+INT(RND(.)*8)*32:C=
345 FORA=CTOB:POKEV+2,A:POKEF+1,AAND10:NE :rem 63
    XT:POKEF+4,
350 POKEF+6,128:POKEF+5,,:POKEF+4,33:POKE :rem 189
    F+1,,:RETURN
400 PRINT"[CLR]{ 6 GIU'}[<2>]{RVS}[<*>] :rem 114
    { 8 DES}{CYN} ":PRINT"[<2>]{RVS}
    { 8 DES}{CYN} { 4 DES}[<6>] ":POKE204
    0,15
410 PRINT"[<2>]{RVS} { 8 DES}{CYN} :rem 39
    { 4 DES}[<6>] ":POKEV+1,101:POKEV+39,
    1:POKEV,32:POKEV+21,1
420 PRINT"[<2>]{RVS}{DES}{YEL}{DES} :rem 151
    [<5>]~{ 3 SPAZI}{DES}{CYN} ~ [<*>]
    [<6>]{OFF}[<*>]{RVS}{ 2 SPAZI}{DES}

```

LOAD

Lightsaver



```

(PUR)~{ 2 SPAZI}{OFF}~{RVS}{DES}{RED}
~{ 2 SPAZI}{<*>}{DES}{BLU} { 2 DES}
{DES}{GRN}~{<*>}{DES}{<3>} ~{<*>}:P
OKEV+29,. :rem 26
430 PRINT"[<2>]{RVS}{DES}{YEL}{DES}
[<5>] { 2 DES}{DES}{CYN}{OFF}~{RVS}
{DES}{DES}{<6>} { 2 DES}{PUR}{OFF}
[<*>]{<@>}{ 2 DES}{RVS}{DES}{RED}
{ 2 DES}{DES}{BLU} { 2 DES}{DES}
{GRN}{OFF}{<P>}{RVS}{DES}{<3>}
{OFF}~{<*>}:POKEV+23,. :rem 185
440 PRINT"[<2>]{RVS}{DES}{YEL}{DES}
[<5>] { 2 DES}{DES}{CYN} { 2 DES}
{DES}{<6>} {DES}~{ 2 DES}{PUR}{OFF}
[<T>]{RVS}{<*>}{DES}{RED} { 2 DES}
{DES}{BLU}{OFF}{<*>}{RVS}{<*>}{OFF}
~{RVS}{DES}{GRN} { 3 DES}{<3>}"
:rem 87
450 PRINT"[<2>]{RVS}{DES}{YEL}{OFF}{<*>}
{DES}{<5>}{<*>}{RVS}{ 3 SPAZI}{DES}
{CYN} { 2 DES}{DES}{<6>}{OFF}{<*>}
{RVS}{OFF}~{RVS}{PUR}~{ 2 SPAZI}
{OFF}~{DES}{RED}{<*>}{RVS}{OFF}~
{RVS}~{ 2 DES}{BLU}{OFF}{<*>}~
{ 2 DES}{GRN}{<*>}{RVS}{OFF}~{DES}
{RVS}{<3>}" :rem 18
460 PRINT"[<2>]{RVS} { 6 DES}{<5>}" :PRIN
T"[<2>]{RVS} { 3 DES}{<5>}{OFF}{<*>}
{RVS}{ 2 SPAZI}{OFF}~{PRINT"[<2>]
{RVS}{ 4 SPAZI}{<*>}{ 3 GIU'}"
:rem 123
470 PRINT"[WHT]{ 4 SPAZI}USARE LE PADDLE
NELLA PORTA N[<V>]1." :RETURN:rem 163
500 QW=49164:RW=49517 :rem 172
510 FORAW=.TO22:BW=PEEK(AW+QW):POKEAW+QW,
PEEK(AW+RW):POKEAW+RW,BW:NEXT
:rem 109
520 IPBW=141:THENPOKE49262,96:POKE49230,63
:POKE49378,52:POKE49379,3:GOTO540
:rem 129

```

```

530 POKE49262,165:POKE49230,14:POKE49378,
64:POKE49379,0:POKE49481,252:RETURN
:rem 199
540 POKE49481,255:RETURN :rem 180
550 IFPEEK(51457)=114:THENGOTO560:rem 123
552 HS$="000000":HU$="C-64":GOSUB590
:rem 170
560 HT$="":FORA=51458TO51467:HT$=HT$+CHR$
(PEEK(A)):NEXT :rem 208
562 HS$=LEFT$(HT$,6):HU$=RIGHT$(HT$,4)
:rem 166
564 RETURN :rem 127
590 HT$=HS$+HU$+" ":FORA=1:TOLEN(HT$):POKE
51457+A,ASC(MID$(HT$,A,1)):NEXT:RETUR
N :rem 179
600 POKE53269,.:IFPEEK(49164)=141:THENGOSU
B500:IFSL=0:THENSYSX:NS$="000000"
:rem 129
605 POKE53272,28:PRINT"{HOME}{ 12 GIU'}
{YEL}{ 5 SPAZI}PREMI <F1> PER INIZIAR
E." :rem 201
607 IFSL=0:THENS�=6:GOSUB700 :rem 202
610 GOSUB202:GOSUB340:POKE254,0:POKE820,2
55:POKE68,0:SYST :rem 231
620 IFP=2:THENP=1:BL=BL+20 :rem 243
630 IFPEEK(68)=1:THENGOTO80 :rem 74
640 BL=872:SYSX:POKE54296,0:POKE53269,0:P
OKE53280,0:GOSUB800:GOTO60 :rem 38
700 RESTORE:B=589:FORA=.TOB:READM:NEXT:FO
RA=.TO10:READM:J=M*8+12288 :rem 252
710 FORB=.TO7:READM:POKEB+J,M:NEXTB,A:RET
URN :rem 132
800 RESTORE:FORA=832TO895:READM:POKEA,M:N
EXT:POKE54273,.:RETURN :rem 98
900 F=54272:FORA=0TO24:POKEF+A,.:NEXT:A$=
"IL PIU' FORTE":POKE646,6:GOSUB990
:rem 75
910 POKE646,7:POKEF+6,240:POKEF+4,17:POKE
F+24,143 :rem 68
920 FORA=1:TOLEN(A$):FORB=.TO200:STEP10:POK
EF+1,B:NEXT:PRINTMID$(A$,A,1){DES};"
:NEXT :rem 85
930 POKEF+24,140:A=1397 :rem 239
940 PRINT"{ 13 GIU'}{CYN}{ 8 SPAZI}USARE
LA PADDLE PER CAMBIARE.{GIU'}"
:rem 251
941 PRINT"USARE FIRE PER MEMORIZZARE."
:rem 127
942 PRINT"{HOME}{ 9 GIU'}{<6>}
{ 13 SPAZI}{<@>}{<@>}{<@>}{<@>}:H
U$="" :rem 193
943 POKEA+40,30:POKEA+F+40,10:POKEF+A,1
:rem 221
945 B=29-PEEK(54297)/9:POKEA,B:POKE646,(P
EEK(646)+1)AND255:GOSUB990 :rem 125
950 POKEF+1,B*8+15:IFPEEK(56321)AND4:THENG
OTO945 :rem 33
955 POKEF+4,129:FORC=0TO90:STEP1.5:POKEF+1
,C:NEXT:POKEF+4,17:POKEA+40,32
:rem 75
960 A=A+2:HU$=HU$+CHR$(B+64):IFA<1405:THEN
943 :rem 231
962 PRINT"{ 4 GIU'}":POKEF+4,0:POKE51457,
114 :rem 81
965 GOSUB590:PRINT"{HOME}":FORA=1:TO13
:rem 183
970 PRINT"{ 32 SPAZI}":NEXT:GOTO90
:rem 198
990 PRINT"{HOME}{ 2 GIU'}{DES}U{ 29 *}~"S
PC(9)"B"SPC(29)"B" :rem 65
995 PRINT"{DES}J{ 29 *}~"K{HOME}{ 3 GIU'}
{ 2 DES};" :RETURN :rem 191

```

LE APPLICAZIONI
DEL COMPUTER
NELL'UFFICIO
MODERNO

LA COMUNICAZIONE
CAMBIATA

UNITA' A DISCHI
per Personal Computer

WORDSTAR

urbanistica
informatica

GRUPPO
EDITORIALE
JACKSON

La biblioteca che fa testo.

ritagliare (o fotocopiare) e spedire in busta chiusa a:
GRUPPO EDITORIALE JACKSON - Divisione Libri - Via Bonellini, 12 - 20124 Milano

Cod. 801P Pag. 224 Lire 30.000

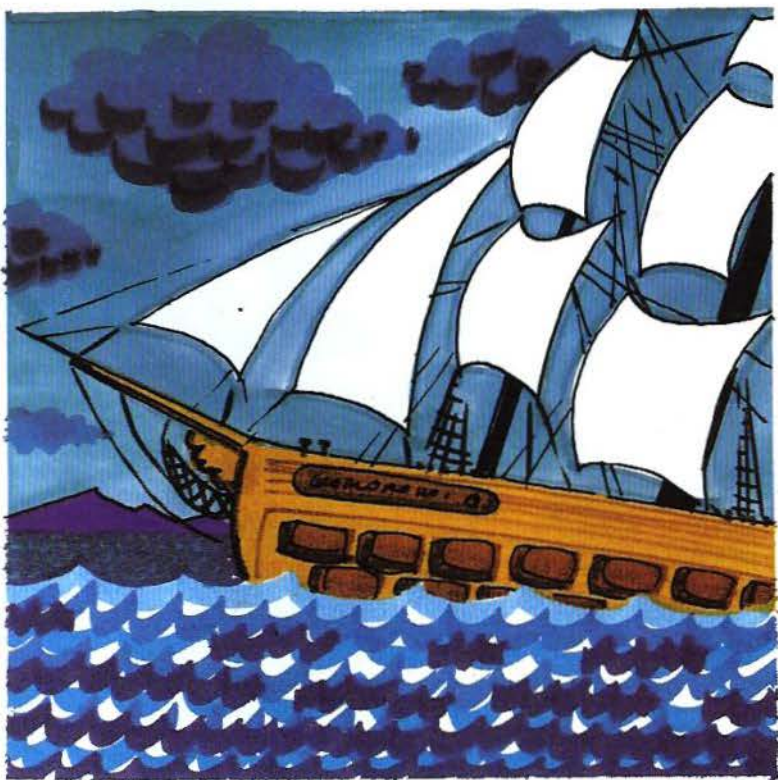
Verso le Indie

di M. J. Winter
trad. e adatt.
di F. Sarcina

Un programma di simulazione storica che ti renderà protagonista di una delle più grandi imprese nella storia delle esplorazioni via mare: arrivare in India attraverso la circumnavigazione dell'Africa. Per C64.

La storia dei programmi di simulazione di eventi storici risale agli albori del personal computer: coloro che hanno avuto a che fare con l'ormai mitico PET 2001 sicuramente si ricorderanno di alcuni di essi. SUPERCOMMODORE ti presenta una versione dovutamente aggiornata di questo tipo di programma.

Lo scenario storico di VERSO LE INDIE è la fine del XV secolo: l'epoca delle grandi scoperte geografiche. Sei un audace capitano di vascello portoghese che, stimolato dalle possibilità di gloria e di altissimi guadagni, decide di battere sul tempo Vasco De Gama, arrivando a trovare una via per l'India attraverso est, partendo da Lisbona e tentando di circumnavigare per la prima volta l'Africa. Durante il viaggio puoi incontrare i medesimi pericoli affrontati dai veri esploratori: fame, sete, navi piratesche, indigeni, tempeste, ammutinamenti e attacchi dei mercanti arabi. La subroutine alla linea 2200 provvede a fornirti le regole del gioco. Alla fine di ogni settimana puoi osservare una mappa con la tua attuale posizione e osservare la situazione completa della tua nave. Le miglia percorse dipendono dal tempo atmosferico incontrato; ogni settimana le tue riserve di cibo, acqua e provviste di materiale (vele, ecc.) diminuiscono di una unità. Se il tuo viaggio dura da più di 30 settimane, anche il morale dei tuoi uomini comincia a scendere al ritmo di una unità ogni 7 giorni. Ogni settimana devi affrontare una nuova avventura; la linea 400 manda il programma all'evento appropriato. Nella prima parte del viaggio incontri balene ed altre navi e affronti terribili tempeste. Anche dopo il passaggio del Capo di Buona Speranza, e l'incontro con un



esperto pilota indiano, puoi sempre essere attaccato in qualsiasi momento dagli arabi.

Se hai navigato abbastanza da visitare le isole Canarie e poi quelle di Capo Verde, le tue scorte e il morale dell'equipaggio migliorano. A volte il capitano di una nave incontrata per caso può sfidarti ad una gara di velocità: se vinci, oltre alla posta in palio ottieni anche un aumento del morale, ma se perdi, i tuoi marinai saranno sempre più scontenti. Quando incontri una balena per la tua strada puoi decidere se tentare di arpionarla oppure proseguire; uccidendola provochi ancora un miglioramento del morale.

Ogni tanto avvisti un estuario di fiume, facile approdo per la tua nave. Se decidi di fermarti, potrai risanare le tue scorte e far svagare i marinai, ma qualche volta verrai visitato da indigeni. Alcuni di essi sono amichevoli e ti offriranno dell'oro in cambio delle solite carabattole, ma certe volte la tribù ti attacca. Devi a questo punto digitare il comando CORRERE seguito da RETURN il più velocemente possibile, per battere in ritirata e salvare la nave. Segui la stessa procedura, se vieni attaccato dai mercanti arabi, gelosi delle loro rotte commerciali; lo stesso Vasco De Gama ebbe numerosi di questi sfortunati incontri.

Importanti routine del programma

440-660 Caccia alle balene
710-1100 Incontro con altre navi
1120-1430 Sbarco in estuario ed incontro con gli indigeni
1150-1700 Incontro con mercanti arabi
1720-1960 Stampa giornale di bordo

2530-2750 Stampa mappa

NOTA BENE: In questo listato il carattere [SHIFT £] appare come segno ~ sottolineato nelle seguenti linee di programma:

130 2610
160 2620
310 2630

320 2640
500 2650
920 2660
2560 2700
2580 2720
2590 2730
2600 2740

Quando digiti il programma presta attenzione a queste linee!

Verso le Indie per C64

```

100 DIMM(7),M$(6):X=RND(-TI) :rem 87
110 FG=1.6:KB=151:HP=72:PRINT "{CLR}":IFPE
EK(1024)=32:THENKB=197:HP=29 :rem 94
120 M$(0)="":M$(1)="FERMATA ALLE ISOLE CA
NARIE":M$(5)="TERRA! E' CALCUTTA!"
:rem 112
130 DH$="{ 2 SPAZI}:[<L>]{GIU'}{ 3 SIN}
[<*>]{RVS}{ 3 SPAZI}:[<*>]{GIU'}
{ 4 SIN}{ 4 SPAZI}{GIU'}{ 4 SIN}
{ 4 SPAZI}{GIU'}{ 5 SIN}{ 3 SPAZI}
{OFF}{GIU'}{ 3 SIN}{<L>}{GIU'}
{ 4 SIN}{<*>}{RVS}{ 6 SPAZI}{OFF}~
:rem 146
140 M$(2)="ISOLE DI CAPO VERDE":M$(3)="DO
PPIATO IL CAPO DI BUONA SPERANZA"
:rem 218
150 M$(4)="RACCOLTO UN PILOTA INDIANO"
:rem 210
160 YSS$="-{GIU'}{SIN}{RVS}+{GIU'}{SIN}
{OFF}{GIU'}{ 2 SIN}{<*>}{RVS}{OFF}
~":HSS$="{RED}{GIU'}{SIN}{RVS}Z{OFF}
{GIU'}{SIN}{GIU'}{ 2 SIN}{<*>}{RVS}
{OFF}~":rem 149
170 MSS$="{DES}"+CHR$(20)+"{GIU'}":MSS$=MSS$
+MSS$:MSS$=MSS$+MSS$ :rem 53
180 WH$(1)="BONACCIA":WH$(2)="MOLTO CALMO"
:WH$(3)="VENTI FAVOREVOLI":rem 12
190 WH$(4)="FORTE VENTO IN POPPA":WH$(5)=
WH$(4):WH$(6)="DEBOLE BURRASCA"
:rem 145
200 M$(1)=50:M$(2)=100:M$(3)=150:M$(4)=20
0:M$(5)=250:M$(6)=300:M$(7)=50
:rem 7
210 GOSUB2530 :rem 221
220 DT$="{ 2 SIN}. {GIU'}{SIN}. { 2 SIN}.
{ 2 SIN}. {GIU'}{SIN}. {GIU'}{SIN}.
{GIU'}{SIN}. {GIU'}{SIN}. { 2 SIN}.
{GIU'}{SIN}. {GIU'}{SIN}. {GIU'}{SIN}.
{GIU'}{SIN}. {GIU'}{SIN}. {GIU'}{SIN}.
{GIU'}{SIN}. {GIU'}{SIN}. {GIU'}{SIN}.
{SIN}." :rem 165
230 DT$=DT$+"{GIU'}{SIN}. {GIU'}{SIN}.
{GIU'}{SIN}. {GIU'}{SIN}. {SU}. {SU}. {SU}.
{SU}. {SU}. {SU}. {SU}. {SU}. {SU}.
{SU}. {SU}. {SU}. {SU}. {SU}. {SU}..
..":rem 165
240 DT$=DT$+"{SU}. {SU}{SIN}." :rem 3
250 DEFFNR(X)=INT(RND(1)*X+1) :rem 178
260 F$="{ 2 SPAZI}<.[< 2 @>]V{ 13 SPAZI}
(<.[< 2 I>]N{ 34 SPAZI}":rem 115
270 F$=F$+"{ 28 SPAZI}":rem 25
280 F$=F$+"<.[< 2 +>]V{ 40 SPAZI}{
[< 2 +>]N{ 15 SPAZI}":rem 20
290 F$=F$+"{ 28 SPAZI}":rem 27
300 D$="{HOME}{ 32 GIU'}":rem 175

```

```

310 S$="{ 3 SPAZI}{<M>}{ 2 SPAZI}{<M>}
{GIU'}{ 5 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 6 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 6 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 4 SIN}{<G>}{<M>}{GIU'}{ 6 SIN}
[<*>]{RVS}{ 3 Z}{OFF}~":rem 136
320 SSS$="{ 3 SPAZI}{<M>}{ 2 SPAZI}{<M>}
{GIU'}{ 5 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 6 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 6 SIN}{< 3 +>}{< 2 +>}{GIU'}
{ 4 SIN}{<G>}{<M>}{GIU'}{ 6 SIN}
[<*>]{RVS}{ 5 SPAZI}{OFF}~":rem 78
330 GOSUB2200 :rem 218
340 ML=8:GOSUB1710:FORWK=1TO52:Z=FRE(0):F
ORI=1TO10:GETAS$NEXT :rem 102
350 GOSUB2040:POKE53281,3 :rem 169
360 GOSUB1710 :rem 226
370 GOSUB1970 :rem 235
380 X=FNR(GG):IFGG=6THENX=2*FNR(4)-1
:rem 168
390 IFML<1200ANDX=4THENX=1 :rem 84
400 ONXGOSUB430,700,430,1110,1440,1540,15
40 :rem 246
410 FD=FD-1:SP=SP-1:WT=WT-1:IFWK>30THENCH
=CH-1 :rem 118
420 NEXT :rem 213
430 : :rem 209
440 DZ=17+INT(8*RND(1)) :rem 10
450 PRINT "{CLR}AVVISTATE DELLE BALENE"
:rem 193
460 PRINT "{GIU'}VUOI TENTARE LA FORTUNA?
(S-N)" :rem 220
470 AS$="{GETAS:IFAS="N"THEN640 :rem 198
480 IFAS<>"S"THEN470 :rem 107
490 PRINT "{CLR}{GIU'}{ 11 SPAZI}W{GIU'}
{SIN}{<+>}~{GIU'}{ 3 SIN}{<+>} V"
:rem 190
500 PRINT "{ 8 SPAZI}{<*>}{RVS}{ 3 SPAZI}
{OFF}~":rem 146
510 PRINT "~{<5>]JK{SPAZI}JKJKJKJ{ 2 K]JKJ
KJKJKJKJKJ[BLK}" :rem 246
520 PRINT "[HOME]PREMI {RVS}H{OFF}"
:rem 228
530 GOSUB670 :rem 181
540 IFPEEK(KB)>HPTHEN530 :rem 4
550 DC=0:PRINTLEFT$(D$,3)TAB(13)" {GIU'}
{SIN}~{GIU'}{SIN}V{GIU'}{SIN}";
:rem 4
560 DC=DC+1:GOSUB670:PRINTLEFT$(D$,3+DC)T
AB(13)" {GIU'}{SIN}~{GIU'}{SIN}V
{GIU'}{SIN}"; :rem 193
570 IFDC<>DZ-6THEN560 :rem 247
580 B$="*":OPEN3,3:INPUT#3,B$:CLOSE3
:rem 65
590 H$=LEFT$(B$,1) :rem 170
600 IFH$="<"ORH$="."ORH$="[">]"ORH$="V"O
RH$="("ORH$="[">]"ORH$="N"THEN630
:rem 170

```

LOAD

Verso
le Indie

```

610 IFH$="[<+>]"ORH$=","THEN630 :rem 59
620 PRINTLEFT$(D$,3+DC)TAB(13)" {GIU'}
{SIN}-{GIU'}{SIN}V{SIN}{SU} {GIU'}
{SIN}-{GIU'}{SIN}V { 2 SU}MANCATO";:G
OTO640 :rem 199
630 PRINT"{ 7 SU}BUON COLPO":FD=FD+2
:rem 250
640 PRINTLEFT$(D$,23)"{ 4 SU}PREMI {RED}R
ETURN{BLK}" :rem 81
650 A$="":GETAS:IFAS<>CHR$(13)THEN650
:rem 40
660 RETURN :rem 124
670 PRINTLEFT$(D$,DZ)LEFT$(F$,39)
:rem 158
680 F$=MID$(F$,2)+LEFT$(F$,1) :rem 238
690 RETURN :rem 127
700 : :rem 209
710 PRINT"{CLR}{RED}"TAB(25)$ :rem 30
720 PRINT"{BLK}AVVISTATA UNA NAVE"
:rem 187
730 PRINT"{RVS}A{OFF}VVICINARSI O {RVS}F
{OFF}UGGIRE" :rem 74
740 A$="":GETAS:IFAS<>"A"ANDAS<>"F"THEN74
0 :rem 51
750 IFAS="A"ANDRND(1)>.2THEN810 :rem 241
760 ES=.5:IFAS="F"THENES=.8:GOTO780
:rem 70
770 PRINT"{GIU'}E' UNA NAVE PIRATA!":PRIN
T"{GIU'}TI VOLTI E FUGGI" :rem 218
780 : :rem 217
790 IFRND(1)>ESTHEN2030 :rem 95
800 GOTO1080 :rem 154
810 FL=0 :rem 152
820 PRINT"{CLR}IL CAPITANO TI SFIDA AD UN
A GARA":PRINT"DI VELOCITA'{GIU'}"
:rem 145
830 IFRND(1)>.5THEN850 :rem 253
840 PRINT"3 PEZZI DEL SUO ORO CONTRO 3 BA
RILI{ 5 SPAZI}DELLE TUE PROVVISI-":G
OTO870 :rem 15
850 FL=1:PRINT"3 BARILI DELLE SUE PROVVIS
TE CONTRO" :rem 136
860 PRINT"3 PEZZI DEL TUO ORO." :rem 31
870 PRINT"{GIU'}ACCETTI? {RVS}S{OFF} O
{RVS}N{OFF}?" :rem 52
880 A$="":GETAS:IFAS="S"THEN930 :rem 210
890 MSS$="{DES}"+CHR$(20)+"{GIU'}":MSS=MSS
+MSS:MSS=MSS+MSS :rem 62
900 IFAS="N"THENRETURN :rem 107
910 GOTO880 :rem 115
920 YS$="{BLK}-{GIU'}{SIN}{RVS}+{GIU'}
{SIN}{OFF}-{GIU'}{ 2 SIN}{<*>}{RVS}
{OFF}~":HS$="{RED} -{GIU'}{SIN}{RVS}
Z{OFF}{GIU'}{SIN}-{GIU'}{ 2 SIN}{<*>]
{RVS}{OFF}~" :rem 41
930 MSS$="{DES}"+CHR$(20)+"{GIU'}":MSS=MSS
+MSS:MSS=MSS+MSS :rem 57
940 PRINT"{CLR}{BLK}"; :rem 204
950 PRINTTAB(36)YS$:PRINT"{ 2 GIU'}"TAB(3
6)HS$ :rem 9
960 FORT=1TO1000:NEXT :rem 37
970 YX=INT(RND(1)*10)+25:HX=INT(RND(1)*9)
+25:IFHX=YXTHENYX=YX+1 :rem 124

```

```

980 MX=YX:W$="LA TUA":IFYX<HXTHENMX=HX:W$
="LA SUA" :rem 148
990 FORJ=1TOMX :rem 137
1000 IFYX<JTHEN1020 :rem 121
1010 PRINT"{HOME}"MSS :rem 106
1020 IFHX<JTHEN1040 :rem 108
1030 PRINT"{HOME}{ 6 GIU'}"MSS :rem 210
1040 NEXTJ :rem 78
1050 PRINT"{BLK}{HOME}{ 15 GIU'}"W$ NAVE
HA VINTO" :rem 59
1060 IFMX=YXTHENGP=GP-(FL=0)*3:SP=SP+3*FL
:CH=CH+2 :rem 30
1070 IFMX=HXTHENGP=GP-FL*3:SP=SP+3*(FL=0)
:CH=CH-2 :rem 16
1080 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 60
1090 A$="":GETAS:IFAS<>CHR$(13)THEN1090
:rem 134
1100 RETURN :rem 162
1110 : :rem 253
1120 IFRND(1)>.7THENRETURN :rem 107
1130 PRINT"{CLR}HAI AVVISTATO UN FIUME"
:rem 213
1140 PRINT"{GIU'}VUOI SBARCAR PER RIFORN
IRTI?" :rem 92
1150 PRINT"{GIU'}{RVS}S{OFF} O {RVS}N
{OFF}" :rem 225
1160 A$="":GETAS:IFAS<>"S"ANDAS<>"N"THEN1
160 :rem 167
1170 IFAS="N"THENCH=CH-2:RETURN :rem 135
1180 PRINT"{ 2 GIU'}SEI APPRODATO E TI SE
I RIFORNITO." :rem 245
1190 IFRND(1)>.5THEN1400 :rem 85
1200 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 54
1210 A$="":GETAS:IFAS<>CHR$(13)THEN1210
:rem 122
1220 PRINT"{CLR}CI SONO INDIGENI
{ 4 SPAZI}O{ 3 SPAZI}O{GIU'}
{ 6 SIN}J{RVS}{RED} {BLK}{OFF}K
[<Z>]{RVS}{GRN} {OFF}{BLK}[<X>]
{GIU'}{ 6 SIN}V{ 3 SPAZI}V" :rem 164
1230 PRINT"{GIU'}{RVS}A{OFF}VVICINARSI OR
{RVS}F{OFF}UGGIRE?" :rem 24
1240 A$="":GETAS:IFAS<>"A"ANDAS<>"F"THEN1
240 :rem 139
1250 IFAS="F"THEN1400 :rem 119
1260 IFRND(1)>.5THEN1290 :rem 90
1270 PRINT"{ 3 GIU'}GLI INDIGENI TI DANNO
DELL'ORO IN CAMBIODI CIANFRUSAGLIE.
" :rem 240
1280 GP=GP+10:CH=CH+1:GOTO1400 :rem 227
1290 PRINT"{CLR}{ 2 SPAZI}O{ 3 SPAZI}O
{GIU'}{ 6 SIN}J{RVS}{YEL} {BLK}{OFF}
KE[<Z>]{RVS}{RED} {OFF}{BLK}[<X>]
{GIU'}{ 6 SIN}V - V"; :rem 143
1300 PRINT"{ 2 SU}{ 5 SPAZI}O{ 3 SPAZI}O
{GIU'}{ 6 SIN}J{RVS}{GRN} {BLK}{OFF}
KE[<Z>]{RVS}[<I>] {BLK}{OFF}[<X>]
{GIU'}{ 6 SIN}V - V" :rem 192
1310 PRINT"{ 3 GIU'}APPAIANO ALTRI INDIGE
NI!":PRINT"{GIU'}CORRI ALLA NAVE!"
:rem 4
1320 PRINT"{GIU'}BATTI {BLU}CORRERE{BLK}
E PREMI RETURN" :rem 127
1330 TI$="000000" :rem 41
1340 INPUTA$:IFAS<>"CORRERE"THEN1340
:rem 178
1350 IFTI<400THEN1370 :rem 191
1360 PRINT"{GIU'}TROPPO LENTO. SEI MORTO.
":GOTO2790 :rem 37

```

```

1370 PRINT{"GIU'}WHEW! HAI SALVATO LA PEL
LE MA PERSO" :rem 11
1380 PRINT"IL CIBO E L'ACQUA." :CH=CH-1
:rem 139
1390 GOTO1410 :rem 204
1400 FD=10:WT=10:CH=CH+1 :rem 89
1410 PRINTLEFTS(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 57
1420 AS="":GETAS:IFAS<>CHRS(13)THEN1420
:rem 128
1430 RETURN :rem 168
1440 : :rem 3
1450 IFRND(1)>.5THENRETURN :rem 111
1460 POKE53281,12 :rem 140
1470 PRINT{"CLR}TEMPESTA TERRIBILE"
:rem 53
1480 IFRND(1)>.9THENPRINT{"GIU'}LA NAVE N
AUFraga E TUTTI MUOIONO." :GOTO2790
:rem 236
1490 PRINT{"GIU'}L'HAI SUPERATA, MA HAI P
ERSO DELLE":PRINT{"GIU'}PROVVISTE IN
MARE" :rem 127
1500 SP=SP-4 :rem 170
1510 PRINTLEFTS(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 58
1520 AS="":GETAS:IFAS<>CHRS(13)THEN1520
:rem 130
1530 POKE53281,3:RETURN :rem 116
1540 : :rem 4
1550 PRINT{"CLR}ACQUE NEMICHE" :rem 151
1560 PRINT{"GIU'}I MERCANTI ARABI TENTERA
NNO DI NON{ 6 SPAZI}FARTI PASSARE"
:rem 164
1570 PRINTLEFTS(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 64
1580 AS="":GETAS:IFAS<>CHRS(13)THEN1580
:rem 142
1590 IFRND(1)>.5THENRETURN :rem 116
1600 PRINT{"CLR}{GIU'}":TAB(8)DH$:PRINT"
{HOME}{GIU'}":TAB(20)DH$ :rem 240
1610 PRINT{"HOME}{ 9 GIU'}AVVISTATE DELLE
GALERE ARABE" :rem 109
1620 PRINT{"GIU'}BATTI {BLU}FUGGIRE{BLK}
E PREMI RETURN" :rem 121
1630 TI$="000000" :rem 44
1640 INPUTAS:IFAS<>"FUGGIRE"THEN1640
:rem 175
1650 IFTI<400THEN1670 :rem 197
1660 PRINT{"GIU'}TI HANNO AFFONDATO." :GOT
O2790 :rem 197
1670 PRINT{"GIU'}IL TUO PILOTA CONDUCE IN
SALVO LA NAVE." :rem 8
1680 PRINTLEFTS(D$,23)"PREMI {RED}RETURN
{BLK}" :rem 66
1690 AS="":GETAS:IFAS<>CHRS(13)THEN1690
:rem 146
1700 RETURN :rem 168
1710 : :rem 3
1720 GOSUB2530:Q=INT(ML/200) :rem 114
1730 IFQ>1THENPRINTLEFTS(D$,5);"{ 9 DES}
";LEFTS(DT$,3*Q) :rem 52
1740 PRINT{"HOME}{BLK}{ 2 SPAZI}GIORNALE
DI BORDO" :rem 148
1750 WK$=RIGHTS("{ 2 SPAZI}"+STR$(WK),4)
:rem 162
1760 ML$=RIGHTS("{ 2 SPAZI}"+STR$(ML),4)
:rem 145
1770 FD$=RIGHTS("{ 2 SPAZI}"+STR$(FD),4)
:rem 116
1780 SP$=RIGHTS("{ 2 SPAZI}"+STR$(SP),4)
:rem 167

```

```

1790 GP$=RIGHTS("{ 2 SPAZI}"+STR$(GP),4)
:rem 144
1800 CH$=RIGHTS("{ 2 SPAZI}"+STR$(CH),4)
:rem 112
1810 WT$=RIGHTS("{ 2 SPAZI}"+STR$(WT),4)
:rem 177
1820 PRINTLEFTS(D$,14)TAB(20)"SETTIMANA
{ 4 SPAZI}";WK$ :rem 102
1830 PRINTTAB(20)"MIGLIA{ 7 SPAZI}";ML$
:rem 210
1840 PRINTTAB(20)"CIBO{ 9 SPAZI}";FD$
:rem 46
1850 PRINTTAB(20)"ACQUA{ 8 SPAZI}";WT$
:rem 158
1860 PRINTTAB(20)"PROVVISTE{ 4 SPAZI}";SP
$ :rem 254
1870 PRINTTAB(20)"ORO{ 10 SPAZI}";GP$
:rem 17
1880 PRINTTAB(20)"MORALE{ 7 SPAZI}";CH$
:rem 214
1890 PRINT{"GIU'}":TAB(14)M$(G) :rem 74
1900 IFG=1ORG=2THENPRINTTAB(14)"PRESI CIB
O ED ACQUA"; :rem 193
1910 IFG=4THENPRINTTAB(14)"A MALINDI"
:rem 222
1920 IFG=5THENPRINTTAB(10){GIU'}{RVS}SEI
IN INDIA!";PRINT{RVS}CONGRATULAZI
ONI!{HOME}"; :rem 254
1930 IFG=5THENPRINT"LA STORIA HA SBAGLIAT
O." :END :rem 210
1940 PRINTLEFTS(D$,24)"PREMI C" :rem 249
1950 AS="":GETAS:IFAS<>"C"THEN1950
:rem 97
1960 PRINT{"CLR}":RETURN :rem 78
1970 : :rem 11
1980 IFFD<1THENPRINT{"CLR}E' FINITO IL CI
BO":PRINT{"GIU'}SEI MORTO DI INEDIA.
" :rem 197
1990 IFSP<1THENPRINT{"CLR}SONO FINITE LE
PROVVISTE":PRINT{"GIU'}SEI MORTO." :G
OTO2790 :rem 103
2000 IFCH<2THENPRINT{"CLR}IL MORALE E' A
TERRA":PRINT{"GIU'}AMMUTINAMENTO: TI
UCCIDONO" :rem 131
2010 IFCH<2THEN2790 :rem 76
2020 RETURN :rem 164
2030 PRINT{" 2 GIU'}MA...":PRINT{"GIU'}I
PIRATI TI ABBORDANO E TI AFFONDANO!"
:GOTO2790 :rem 247
2040 POKE53281,7 :rem 91
2050 WH=FNR(7):G=0:GG=5:CM=M$(WH)*FG
:rem 89
2060 PRINT{"CLR}{ 6 GIU'}TEMPO: "
:rem 140
2070 IFWH=7THEN2160 :rem 99
2080 PRINTWH$(WH):IFWH<3THENCH=CH-1
:rem 193
2090 IFML<800ANDML+CM>800THENG=1:WT=10:FD
=FD+3:SP=SP+6:IFFD<10THENFD=10
:rem 121
2100 IFML<1500ANDML+CM>1500THENG=2:WT=10:
FD=FD+3:SP=SP+6:IFFD<10THENFD=10
:rem 206
2110 IFML<5000ANDML+CM>5000THENG=3
:rem 253
2120 IFML<6600ANDML+CM>6600THENG=4
:rem 13
2130 IFML>6600THENG=6 :rem 40
2140 ML=ML+CM:Q=INT(ML/2+.5):IFML>9000THE
NG=5 :rem 206
2150 GOTO2170 :rem 203

```

LOAD

Verso
le Indie



```

2160 PRINT"PIOGGIA INSISTENTE":PRINT"
      {GIU'}RIEMPI I SERBATOI DELL'ACQUA":
      WT=10                                :rem 90
2170 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
      {BLK}"                               :rem 61
2180 A$="":GETA$:IFA$<>CHR$(13)THEN2180
                                           :rem 136
2190 RETURN                               :rem 172
2200 POKE53281,13                         :rem 134
2210 PRINT"{CLR}ENRICO IL NAVIGATORE, PRI
      NCIPPE DI"                           :rem 129
2220 PRINT"{GIU'}PORTOGALLO, CREDE CHE VI
      SIA UNA VIA"                         :rem 74
2230 PRINT"{GIU'}MARITTIMA PER L'INDIA, E
      D OFFRE UN"                          :rem 218
2240 PRINT"{GIU'}PREMIO A CHI LA TROVA. V
      ASCO DE GAMA"                       :rem 237
2250 PRINT"{GIU'}VUOLE PROVARCIL, ED E' QU
      ASI PRONTO A"                       :rem 99
2260 PRINT"{GIU'}PARTIRE. MA LA TUA NAVE
      PUO' SALPARE"                       :rem 67
2270 PRINT"{GIU'}OGGI!":PRINT"{ 3 GIU'}DE
      CIDI DI SFIDARE LA SORTE."          :rem 150
2280 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
      {BLK}"                               :rem 63
2290 A$="":GETA$:IFA$<>CHR$(13)THEN2290
                                           :rem 140
2300 WT=10:GP=20                          :rem 126
2310 PRINT"{CLR}ALTRI HANNO TENTATO. ALCU
      NI MORIRONO"                       :rem 152
2320 PRINT"{GIU'}NELLE TEMPESTE, O DI FAM
      E E DI SETE;"                       :rem 223
2330 PRINT"{GIU'}ALTRI FURONO UCCISI DA P
      IRATI O DA"                         :rem 225
2340 PRINT"{GIU'}INDIGENI. FREQUENTI FURO
      NO GLI"                             :rem 136
2350 PRINT"{GIU'}AMMUTINAMENTI, MOLTI PER
      IRONO UCCISI"                       :rem 100
2360 PRINT"{GIU'}DAI MERCANTI ARABI. TUTT
      O CIO' POTRA'"                     :rem 170

```

```

2370 PRINT"{GIU'}ACCADERTI."             :rem 92
2380 PRINT"{GIU'}{RVS}UN ULTIMO AVVISO
      {OFF}: NON TUTTE LE NAVI"          :rem 34
2390 PRINT"{GIU'}SONO PIRATE. GLI INDIGEN
      I POSSONO"                          :rem 54
2400 PRINT"{GIU'}ESSERTI AMICI. CIBO FRES
      CO,TEMPO BUONO"                    :rem 107
2410 PRINT"{GIU'}E UN ABBONDANTE CARICO D
      'ORO TERRANNO"                     :rem 212
2420 PRINT"{GIU'}ALTO IL MORALE."        :rem 93
2430 PRINTTAB(27)"PREMI {RED}RETURN{BLK}"
                                           :rem 52
2440 A$="":GETA$:IFA$<>CHR$(13)THEN2440
                                           :rem 134
2450 PRINT"{CLR}{GIU'}"TAB(30)SS$
                                           :rem 149
2460 PRINT"{ 3 GIU'}PREMI {RVS}L{OFF} PER
      SALPARE DA LISBONA"                 :rem 185
2470 A$="":GETA$:IFA$<>"L"THEN2470
                                           :rem 102
2480 FORI=1TO30:PRINT"{HOME}"MS$:PRINT"
      {SU}"MS$:NEXT                      :rem 45
2490 SP=50:CH=10:FD=10                   :rem 217
2500 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
      {BLK}"                               :rem 58
2510 A$="":GETA$:IFA$<>CHR$(13)THEN2510
                                           :rem 130
2520 RETURN                               :rem 169
2530 POKE53281,3:PRINT"{CLR}":IFML=0THENP
      RINT"{CLR}{BLK} VERSO LE INDIE":POKE
      53280,3                             :rem 62
2540 PRINT"{BLK}{ 36 R}{YEL}"            :rem 82
2550 PRINTTAB(8)"{RVS}{ 10 SPAZI}{OFF}
      { 7 SPAZI}{RVS}{ 7 SPAZI}"         :rem 179
2560 PRINTTAB(8)"{RVS}{ 2 SPAZI}{OFF}
      { 3 SPAZI}{< 2 J>}{RVS}{ 19 SPAZI}
      "                                    :rem 199
2570 PRINTTAB(8)"{< 2 U>}{ 3 SPAZI}{<C>}
      { 2 SPAZI}{RVS}{ 23 SPAZI}"         :rem 61
2580 PRINTTAB(4)"{<B>}{ 3 SPAZI}{RVS}
      { 3 SPAZI}{OFF}{ 4 SPAZI}{RVS}
      { 19 SPAZI}"                       :rem 26
2590 PRINTTAB(7)"{RVS}{ 8 SPAZI}{OFF}
      [<*>]{RVS}{ 12 SPAZI}{OFF}"        :rem 121
2600 PRINTTAB(6)"{RVS}{ 9 SPAZI}{OFF}
      [<*>]{RVS}{ 2 SPAZI}{<*>}{OFF}
      { 2 SPAZI}{<*>}{RVS}{ 4 SPAZI}{OFF}
      "                                    :rem 210
2610 PRINTTAB(3)"{<B>}{ 2 SPAZI}{RVS}
      { 10 SPAZI}{<*>}{OFF} [<*>]{RVS}
      { 2 SPAZI}{<*>}{OFF}{ 2 SPAZI}{RVS}
      { 3 SPAZI}{OFF}"                  :rem 230
2620 PRINTTAB(6)"{RVS}{ 11 SPAZI}{<*>}
      {OFF} [<*>]{RVS}{OFF}{ 2 SPAZI}
      {RVS}{ 2 SPAZI}{OFF}"             :rem 245
2630 PRINTTAB(6)"{RVS}{ 12 SPAZI}{<*>}
      {OFF} { 3 SPAZI}{<*>}{RVS} "
                                           :rem 23
2640 PRINTTAB(6)"{<*>}{RVS}{ 12 SPAZI}
      {OFF}{ 5 SPAZI}{<*>}"             :rem 6
2650 PRINTTAB(10)"{<*>}{RVS}{ 7 SPAZI}
      {OFF}"                             :rem 83
2660 PRINTTAB(11)"{RVS}{ 6 SPAZI}{OFF}
      "                                    :rem 118
2670 PRINTTAB(11)"{RVS}{ 6 SPAZI}{OFF}"
                                           :rem 206
2680 PRINTTAB(11)"{RVS}{ 6 SPAZI}{OFF}"
                                           :rem 207
2690 PRINTTAB(11)"{RVS}{ 6 SPAZI}{OFF}"
                                           :rem 208

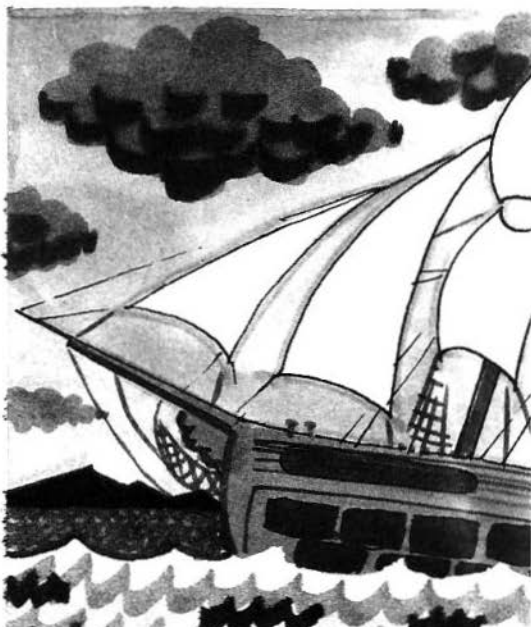
```

```

2700 PRINTTAB(11)"{RVS}{ 5 SPAZI}{OFF}
      ~" :rem 113
2710 PRINTTAB(11)"{RVS}{ 5 SPAZI}{OFF}
      {RVS}{<H>}" :rem 143
2720 PRINTTAB(11)"{RVS}{ 4 SPAZI}{OFF}
      ~ {RVS} " :rem 133
2730 PRINTTAB(11)"{RVS}{ 3 SPAZI}{OFF}
      ~{ 2 SPAZI}{RVS} " :rem 134
2740 PRINTTAB(11)"{<*>}{RVS}{OFF}~"
      :rem 84
2750 IFML>0THENPRINT"{BLK}";:RETURN
      :rem 73
2760 PRINTTAB(20)"{GIU'}{BLK}PREMI {RVS}I
      {OFF} PER INIZIARE"; :rem 178
2770 AS="":GETAS:IFAS<>"I"THEN2770
      :rem 105
2780 RETURN :rem 177
2790 PRINTLEFT$(D$,23)"PREMI {RED}RETURN
      {BLK}" :rem 69
2800 AS="":GETAS:IFAS<>CHR$(13)THEN2800
      :rem 134
2810 PRINT"{CLR}{ 2 GIU'}20 MAGGIO 1498":
      PRINT"{GIU'}VASCO DE GAMA HA RAGGIUN
      TO CALCUTTA" :rem 254
2820 PRINT"{GIU'}SULLA COSTA OCCIDENTALE
      DELL'INDIA," :rem 157
2830 PRINT"{GIU'}DOPO UN VIAGGIO DI 11 ME
      SI E" :rem 236
2840 PRINT"{GIU'}9000 MIGLIA DI NAVIGAZIO
      NE.{ 4 GIU'}" :rem 102
2850 PRINT"{GIU'}GIOCHI ANCORA? {RVS}S
      {OFF} O {RVS}N{OFF}" :rem 143
2860 AS="":GETAS:IFAS="S"THENRUN:rem 91
2870 IFAS<>"N"THEN2860 :rem 208

```

```
2880 PRINT"{CLR}TI AUGURO DI AVERE MIGLIO  
R FORTUNA{ 6 SPAZI}UN'ALTRA  
{ 2 SPAZI}VOLTA!"           :rem 189  
2890 END                       :rem 170
```



IN EDICOLA



Una pubblicazione firmata...



GRUPPO EDITORIALE JACKSON

20124 MILANO - Via Rosellini, 12
Tel. 02/6880451/2/3/4/5

Il Jacksoniano sceglie tra 14 top...



Jackson: una grande, esauriente scelta di periodici per saper tutto ciò che è indispensabile su elettronica, informatica e personal computer. **Videogiochi**, la guida indiscussa al fantastico mondo dei videogames; **Home Computer**, la rivista del computer in casa; **Personal O**, la rivista dedicata al software dei personal computer; **Bit**, la prima rivista europea di personal computer, software, accessori, la più prestigiosa e più diffusa in Italia; **Informatica Oggi**, il punto di riferimento obbligato per chi si occupa di sistemi EDP e di Office Automation; **PC Magazine**, la prima rivista italiana dei sistemi MS-DOS, Personal Computer IBM e compatibili;

Personal O, la rivista indipendente per gli utenti di PC Olivetti; **Compuscuela**, la rivista di informatica nella didattica, per la scuola italiana; **Telecomunicazioni Oggi**, la rivista di telecomunicazioni e telematica; **Automazione Oggi**, il mensile della nuova automazione industriale; **Elettronica Oggi**, la più autorevole rivista di elettronica professionale, strumentazione e componenti; **L'Elettronica**, il quindicinale di politica industriale, componentistica, informatica e telecomunicazioni; **Elektor**, la più diffusa rivista europea di applicazioni e progettazione elettronica; **Strumenti musicali**, il periodico di strumenti musicali e computer-music.

Per sottoscrivere abbonamenti compilare il tagliando e inviare in busta chiusa a:

Gruppo Editoriale Jackson S.r.l. - Via Rosellini, 12 - 20124 Milano, allegando assegno o fotocopia della ricevuta di versamento con vaglia postale intestata a GRUPPO EDITORIALE JACKSON - MILANO.

☐ Sì, desidero sottoscrivere un abbonamento a:

- ☐ Videogiochi (11 n.) L. 30.000 anziché L. 38.500
- ☐ Home Computer (11 n.) L. 31.500 anziché L. 38.500
- ☐ Personal Software (11 n.) L. 34.000 anziché L. 44.000
- ☐ Bit (11 n.) L. 43.000 anziché L. 55.000
- ☐ Informatica Oggi (11 n.) L. 31.000 anziché L. 38.500
- ☐ PC Magazine (10 n.) L. 40.000 anziché L. 50.000
- ☐ Personal O (10 n.) L. 35.000 anziché L. 40.000

- ☐ Compuscuela (9 n.) L. 15.000 anziché L. 18.000
 - ☐ Telecomunicazioni Oggi (10 n.) L. 28.000 anziché L. 35.000
 - ☐ Automazione Oggi (11 n.) L. 30.500 anziché L. 38.500
 - ☐ Elettronica Oggi (11 n.) L. 36.000 anziché L. 44.000
 - ☐ L'Elettronica (22 n.) L. 44.000 anziché L. 55.000
 - ☐ Elektor (12 n.) L. 29.000 anziché L. 36.000
 - ☐ Strumenti Musicali (10 n.) L. 24.000 anziché L. 30.000
- Attenzione per abbonamento all'estero le tariffe devono essere aumentate del 50%

E c'è un super-risparmio a chi si abbona a due o più riviste.

Tutti coloro che sottoscrivono l'abbonamento a due o più riviste godono di un prezzo ulteriormente agevolato, come appare nella seguente tabellina.

Esempio: Bit+Informatica Oggi L. 43.000+31.000=74.000 meno L. 2.000=L. 72.000

Abbonamento	
a 2 riviste L. 2.000 in meno sulla somma dei 2 prezzi d'abbonamento	a 8 riviste L. 20.000 in meno sulla somma degli 8 prezzi d'abbonamento
a 3 riviste L. 4.000 in meno sulla somma dei 3 prezzi d'abbonamento	a 9 riviste L. 25.000 in meno sulla somma dei 9 prezzi d'abbonamento
a 4 riviste L. 7.000 in meno sulla somma dei 4 prezzi d'abbonamento	a 10 riviste L. 30.000 in meno sulla somma dei 10 prezzi d'abbonamento
a 5 riviste L. 10.000 in meno sulla somma dei 5 prezzi d'abbonamento	a 11 riviste L. 35.000 in meno sulla somma degli 11 prezzi d'abbonamento
a 6 riviste L. 13.000 in meno sulla somma dei 6 prezzi d'abbonamento	a 12 riviste L. 40.000 in meno sulla somma dei 12 prezzi d'abbonamento
a 7 riviste L. 16.000 in meno sulla somma dei 7 prezzi d'abbonamento	a 13 riviste L. 44.500 in meno sulla somma dei 13 prezzi d'abbonamento
	a 14 riviste L. 50.000 in meno sulla somma dei 14 prezzi d'abbonamento

- ☐ Allego assegno n° _____ della Banca _____
- ☐ Ho effettuato versamento sul c/c postale n. 11666203 intestato a Jackson - Milano di cui allego fotocopia della ricevuta
- ☐ Vaglia postale e allego fotocopia ricevuta

Nome _____
 Cognome _____
 Azienda _____
 C.A.P. _____ Città _____
 Via _____



**GRUPPO
EDITORIALE
JACKSON**
 San Francisco-Londra-Milano

Caccia al tesoro per VIC 20

di S. Albarelli

Siamo due ragazzi possessori di VIC 20 ed abbiamo elaborato questo gioco. Questo programma gira anche su VIC inespansi, ma attenzione... utilizza la maggior parte della memoria disponibile.

Digitando il "RUN" viene richiesta la difficoltà (1-4).

Dopo di ciò appare sullo schermo il labirinto.

Quando questo ha finito di formarsi in alto a sinistra compare la pedina del giocatore che viene mossa mediante il joystick o la tastiera (con le modifiche indicate).

Lo scopo del gioco è seguire i passaggi, evitare i muri viola, le frecce (solo nelle difficoltà "3" e "4"), che tolgono 1000 punti, e raccogliere i tesori posti sul percorso.

Questi sono rappresentati dai seguenti simboli e valgono:

Puntini = 10 punti
Asterischi = 50 punti
Pi greco = 100 punti
Cuori = 1000 punti

Per uscire dai posti più intricati del labirinto ci sono due "u negative" che, sottraendo 100 punti, portano in un altro luogo la pedina quando vengono sormontate.

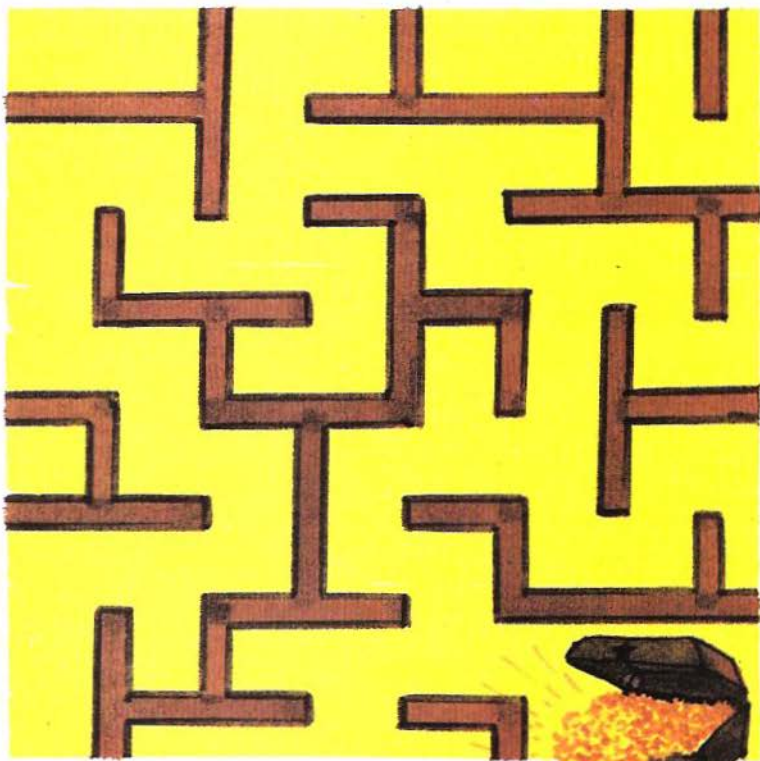
Quando si sono raccolti tutti i tesori presenti nel labirinto, se si è superato il punteggio di 13000 punti, e si torna nel luogo più in alto a sinistra del labirinto, appare il messaggio "MISSIONE COMPIUTA" con il tempo impiegato e, premendo un tasto, si passa al livello superiore di difficoltà.

Se si perde la partita, bisogna premere il tasto rosso sul joystick per ricominciare una nuova caccia al tesoro.

Questo gioco è molto ricco di colori e di effetti sonori.

Ricordiamo che nel labirinto c'è anche una "pedina disturbatrice", che passa laddove non ci sono tesori e che può anche "mangiarsi". Si possono commettere fino a 5 errori.

Noi consigliamo questo gioco ai "PAC-patiti".



Se si volesse variare la velocità della pedina del giocatore, bisogna apportare alla riga 1083 queste modifiche: per aumentare la velocità della pedina bisogna diminuire il numero X alla riga 1083 (FORT=1TOX:NEXTT) per diminuire la velocità della pedina bisogna aumentare il numero X alla riga 1083

Caccia al tesoro da tastiera

In caso non si fosse in possesso del joystick, per poter giocare da tastiera bisogna apportare le seguenti modifiche al programma:

Riga

```
15  cambiare in IFC=6THENGES-
    TAS:IFA$=" " THENRUN
16  cambiare in IFC=6THEN15
1083 cambiare in REM
1084 cambiare in GETA$
1086 cambiare in IFA$="P"THEN-
    POKEY, 32:Y=Y-22
1087 cambiare in IFA$="L"THEN-
    POKEY, 32:Y=Y-1
1088 cambiare in IFA$=";"THEN-
    POKEY, 32:Y=Y+1
1089 cambiare in IFA$="."THEN-
    POKEY, 32:Y=Y+22
```

Così il programma è pronto per essere

usato mediante la tastiera.
I quattro tasti da usare per muovere la
pedina sono:

per andare su "P"
per andare a sinistra "L"
per andare a destra "R"
per andare giù "D"
per ripartire tasto "SPACE"
(barra di spaziatura)

VI AUGURIAMO UN BUON DIVERTI-
MENTO!

Caccia al tesoro per VIC 20

```

0 VA=36879:POKEVA,143:PRINT"(CLR)";
  {GIU'}{ 2 DES}{BLU}*CACCIA AL TESORO*":
  VO=37154 :rem 209
1 PRINT"{PUR}{GIU'}*{YEL}BY STEFANO ALBAR
  ELLI{PUR}*": :rem 112
2 PT=0:C=0:F=7680+(12+(22*12)) :rem 11
3 POKEY,32:Y=7680+67:C=C+1 :rem 226
4 IFC=1THENPOKEVO,255:INPUT"(WHT)DIFFICOL
  TA'(1/2/3/4)";A :rem 114
5 POKEVO,127:IFC=1THENPRINT"(CLR)";
  :rem 23
10 PRINT"{HOME}{WHT}SCORE = "PT;"{HOME}
  { 16 DES}PED"; :rem 29
11 PRINTC; :rem 109
14 IFC=6THENPRINT"{HOME}{GIU'}{ 7 DES}
  {RVS}{PUR}GAME OVER{OFF}":PRINT"PREMI
  {DES}FIRE"; :rem 53
15 IFC=6THENIFJS=341THENRUN :rem 255
16 JS=PEEK(37137)+PEEK(37152):IFC=6THEN15
  :rem 115
18 POKEVA,56 :rem 198
19 IFC>1THEN1000 :rem 155
20 READS :rem 209
21 IFS=10THEN900 :rem 171
22 IFS=0THENPRINT". "; :rem 28
24 IFA>3ANDS=1THENPRINT"{BLU}E{WHT}";
  :rem 248
25 IFA<4ANDS=1THENPRINT"{PUR}{RVS} {OFF}
  {WHT}"; :rem 187
26 IFA=1ANDS=2THENPRINT". "; :rem 164
27 IFA=2ANDS=2THENPRINT"{PUR}{RVS} {OFF}
  {WHT}"; :rem 189
28 IFA>2ANDS=2THENPRINT"{BLU}E{WHT}";
  :rem 252
35 IFS=3THENPRINT"*"; :rem 31
36 IFS=4THENPRINT"[<E>"; :rem 246
37 IFS=5THENPRINT"{RED}S{WHT}"; :rem 237
38 IFS=6THENPRINT"{BLK}{RVS}U{OFF}{WHT}";
  :rem 137
39 IFS=7THENPRINT" "; :rem 253
40 GOTO20 :rem 255
50 DATA1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
  1,1,1,1,1 :rem 81
60 DATA0,0,0,4,0,0,1,2,4,0,1,2,2,0,0,0,
  0,1,1,4,0 :rem 84
70 DATA0,0,0,1,1,3,1,1,3,1,0,0,0,0,1,1,
  0,0,0,1,0 :rem 76
80 DATA0,0,3,2,2,0,0,0,1,0,1,0,2,2,2,2,
  1,1,0,0,0 :rem 83
90 DATA0,1,0,1,1,1,1,0,1,3,0,0,2,0,0,2,
  2,1,0,1,1 :rem 81
100 DATA0,1,0,0,0,0,0,1,0,1,1,1,0,0,0,2,0,0,
  2,3,0,0,0 :rem 115
110 DATA0,1,1,1,1,0,1,0,1,2,1,2,2,2,2,2,0,
  2,1,0,1,4 :rem 131

```

LOAD



Caccia al tesoro
per VIC 20

```

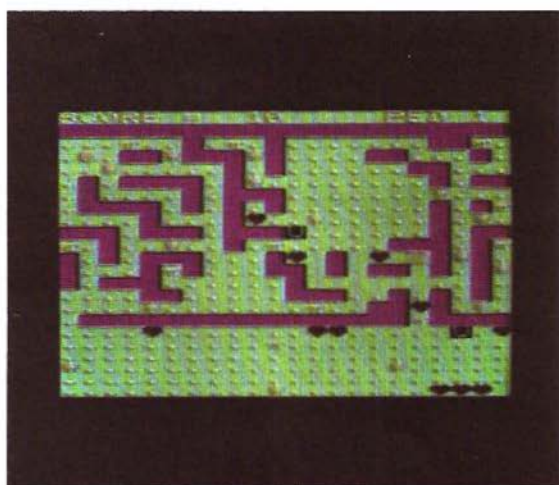
120 DATA0,0,0,0,1,0,0,0,1,5,0,0,3,2,0,0,0,
  2,1,0,3,0 :rem 123
130 DATA1,1,1,0,1,1,1,0,1,1,1,6,0,0,0,2,2,
  2,1,0,1,1 :rem 130
140 DATA4,0,1,0,0,3,0,0,1,0,4,2,2,2,2,2,1,
  1,1,0,1,4 :rem 138
150 DATA1,0,1,0,1,1,1,0,2,0,1,5,7,1,7,5,7,
  7,1,3,1,0 :rem 160
160 DATA1,0,0,0,1,4,1,0,2,0,1,1,7,1,7,1,1,
  7,1,0,1,3 :rem 149
170 DATA1,2,1,1,1,0,0,0,2,0,0,1,7,7,7,7,7,
  7,0,0,1,0 :rem 162
180 DATA2,2,2,2,1,1,0,2,2,2,0,1,1,1,0,2,1,
  2,1,0,1,0 :rem 137
190 DATA0,0,0,0,3,0,0,0,0,0,0,0,4,0,0,1,1,
  5,1,0,3,0 :rem 130
200 DATA0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,
  4,1,1,1,1 :rem 128
210 DATA0,0,0,2,5,2,2,2,2,2,2,2,5,5,0,0,0,
  0,0,6,2,5 :rem 149
220 DATA2,2,0,2,0,0,0,0,0,0,4,2,0,2,2,2,2,
  2,0,0,0,0 :rem 128
230 DATA0,0,0,2,2,2,2,2,0,2,0,2,0,0,0,0,0,
  2,2,2,2,0 :rem 129
240 DATA3,2,2,2,0,0,0,2,0,2,0,2,2,2,2,3,
  2,0,0,0,0 :rem 136
250 DATA0,0,0,0,2,0,2,0,2,0,2,0,2,0,0,2,0,
  2,0,2,2,2 :rem 127
260 DATA2,2,2,2,2,0,0,0,2,3,0,0,2,0,0,0,0,
  2,5,5,5,10 :rem 195
900 TIS="000000" :rem 251
1000 IFVAL(TIS)=>20THEN1003 :rem 21
1002 GOTO1083 :rem 200
1003 B=INT(RND(1)*4)+1 :rem 159
1004 IFB=1ANDPEEK(F-22)=32THEN1009 :rem 191
1005 IFB=2ANDPEEK(F-1)=32THEN1009 :rem 142
1006 IFB=3ANDPEEK(F+1)=32THEN1009 :rem 142
1007 IFB=4ANDPEEK(F+22)=32THEN1009 :rem 195
1008 GOTO1083 :rem 206
1009 IFB=1THENPOKEF,32:F=F-22 :rem 210
1010 IFB=2THENPOKEF,32:F=F-1 :rem 152
1011 IFB=3THENPOKEF,32:F=F+1 :rem 152
1012 IFB=4THENPOKEF,32:F=F+22 :rem 205
1013 IFPEEK(F-22)=81THEN1100 :rem 56
1014 IFPEEK(F-1)=81THEN1100 :rem 6
1015 IFPEEK(F+1)=81THEN1100 :rem 5
1016 IFPEEK(F+22)=81THEN1100 :rem 57
1017 POKEF,102 :rem 253
1083 FORT=1TO200:NEXTT :rem 119
1084 JS=PEEK(37137)+PEEK(37152) :rem 197
1085 IFY=7680+44ANDPT=>13000THEN1600 :rem 55
1086 IFJS=369THENPOKEY,32:Y=Y-22 :rem 220
1087 IFJS=357THENPOKEY,32:Y=Y-1 :rem 167

```

```

1088 IFJS=245THENPOKEY,32:Y=Y+1 :rem 162
1089 IFJS=365THENPOKEY,32:Y=Y+22:rem 217
1090 IFPEEK(Y)=160THEN100 :rem 237
1091 IFPEEK(Y)=30THENPT=PT-1000:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1100 :rem 55
1092 IFPEEK(Y)=46 THENPT=PT+10:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1200 :rem 222
1093 IFPEEK(Y)=42THENPT=PT+50:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1300 :rem 224
1094 IFPEEK(Y)=94THENPT=PT+100:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1400 :rem 21
1095 IFPEEK(Y)=83THENPT=PT+1000:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1500 :rem 69
1096 IFY=7889THENY=Y+21:PT=PT-100:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1500 :rem 206
1097 IFY=8073THENY=Y-44:PT=PT-100:PRINT"
{HOME}SCORE = "PT;"{SIN}{ 2 SPAZI}":
GOTO1500 :rem 200
1099 POKEY,81:GOTO1000 :rem 36
1100 POKE36877,220 :rem 192
1110 FORL=15TO0STEP-1 :rem 6
1120 POKE36878,L :rem 123
1130 FORM=1TO100:NEXTM :rem 97
1140 NEXTL :rem 81
1150 POKE36878,0 :rem 98
1160 POKE36877,0 :rem 98
1170 POKEY,32:GOTO3 :rem 136
1200 POKE36878,5 :rem 99
1210 POKE36877,220 :rem 194
1220 FORT=1TO30:NEXT :rem 237
1230 POKE36877,0 :rem 96
1240 POKE36878,0 :rem 98
1250 GOTO1099 :rem 212
1300 POKE36878,3 :rem 98
1305 POKE36875,225:FORT=1TO250:NEXT :rem 96
1310 POKE36875,235:FORT=1TO250:NEXT :rem 93
1315 POKE36875,225:FORT=1TO250:NEXT :rem 97
1325 POKE36875,0 :rem 99
1330 POKE36878,0 :rem 98
1340 GOTO1099 :rem 212
1400 POKE36878,5 :rem 101
1402 FORH=1TO2 :rem 57
1405 FORL=254TO240+INT(RND(1)*10)STEP-1 :rem 0
1410 POKE36876,L :rem 123
1415 NEXTL :rem 86
1420 POKE36876,0 :rem 96
1425 FORT=1TO250:NEXTT :rem 124
1430 NEXTH :rem 79
1435 POKE36878,0 :rem 104
1440 GOTO1099 :rem 213
1500 POKE36878,10 :rem 146
1505 FORL=225TO240 :rem 13
1510 POKE36875,L :rem 123
1512 FORT=1TO10:NEXTT :rem 67
1515 NEXTL :rem 87
1520 POKE36875,0 :rem 96
1525 POKE36878,0 :rem 104
1530 GOTO1099 :rem 213
1600 PRINT"{CLR}";POKE36878,10:FORT=128T
O250:POKE36875,L:NEXTL:POKE36875,0 :rem 112
1610 POKEVA,143 :rem 82

```



```

1620 PRINT"{GIU'}{HOME}{PUR}@*{@*{YEL}BR
AVISSIMO{PUR}@*{@*@" :rem 15
1630 PRINT"{GIU'}{BLU}{DES}HAI{DES}COMPIU
TO{ 2 DES}LA{DES}TUA":PRINT"{GIU'}
{BLK}{ 6 DES}*MISSIONE*" :rem 42
1640 PRINT"{YEL}IN{DES}";MID$(TIS,3,2);"
MINUTI" :rem 75
1650 PRINT"{YEL}E{ 2 SPAZI}";RIGHT$(TIS,2
);" SECONDI" :rem 81
1660 PRINT"{GIU'}{YEL}@*{@*{BLU}PREMI{DES}
UN{DES}TASTO{YEL}*{@*@" :rem 210
1670 GETAS:IFAS=""THEN1670 :rem 191
1680 RUN :rem 196

```





Supertank per C64

di **B. Litinsky**

trad. e adatt.

di **S. Albarelli** e **M. Anticoli**

In questo inusuale gioco per C64 lo scopo è di combattere i tank avversari che sono penetrati nel vostro territorio.

Contemporaneamente la vostra meta è di raggiungere un grado militare più elevato.

Gli ordini del comando di stato maggiore sono di intercettare e distruggere il maggior numero di carriarmati nemici che hanno invaso il paese. Ma purtroppo non è possibile riconoscere i carriarmati nemici da quelli alleati, che non recheranno alcun male ad un loro connazionale.

Il gioco

Facendo girare il programma si ascolterà l'inno militare americano. Subito dopo sarà possibile scegliere tra uno dei tre carriarmati a disposizione, dei quali verrà mostrato il curriculum:

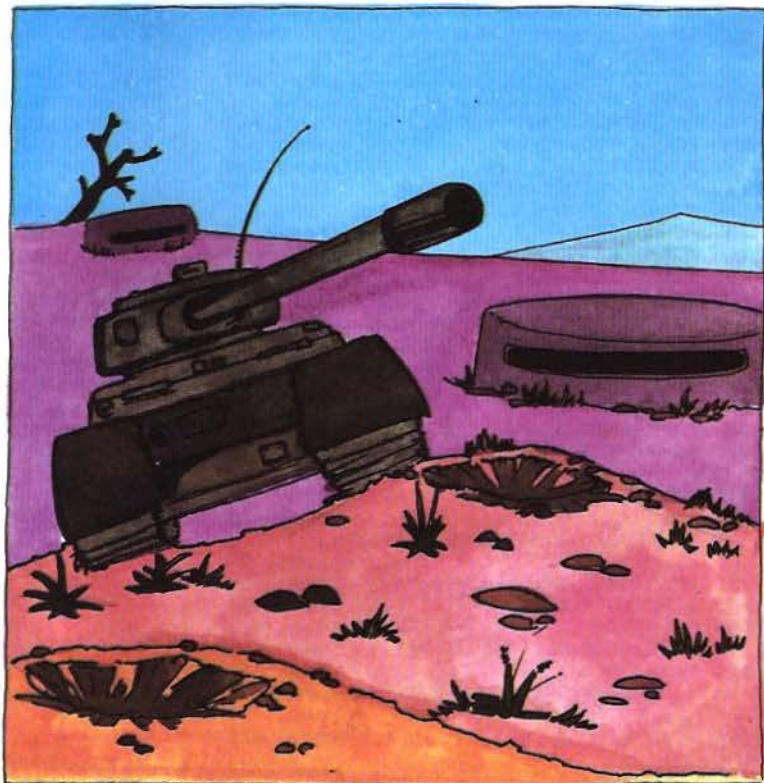
	Alta	Media	Bassa
1. Tiger	Res.	Fuoco	Vel.
2. T-34	Vel.	Res.	Fuoco
3. Sherman	Fuoco	Vel.	Res.

In questa tabella la parola "vel." sta per "velocità" e "res." per resistenza. Quest'ultima indica il numero di colpi a cui può resistere il carroarmato prescelto e, che corrispondono a:

- 5 proiettili, se la resistenza è alta;
- 3 proiettili, se la resistenza è media
- 1 proiettile, se la resistenza è bassa.

Controllo del tank

Ogni carroarmato nemico si muove sullo schermo da sinistra a destra.



Utilizzando il joystick per muovere il mirino del proprio carroarmato è possibile mirare e cercare di colpire il tank avversario; premendo il pulsante di fuoco si vedrà volare la bomba, che partirà dalla sinistra dello schermo fino ad arrivare nel centro del mirino.

Naturalmente, mentre essa compie la sua parabola il tank nemico tenta di fuggire, aumentando anche la velocità, e se non viene centrato, procederà un po' e poi scaricherà un colpo sul carroarmato americano, il quale, perciò, perderà un grado di resistenza.

Se si riuscirà a colpire il tank nemico,

esso esploderà, si acquisteranno 10 punti e si cambierà il territorio di gioco.

Punteggi finali e gradi

Al termine della partita verrà mostrato il grado raggiunto, che viene assegnato a seconda del punteggio.

Se non si totalizzerà alcun punteggio, si verrà banditi come una "burba". Ma non si deve disperare, perché sarà possibile ritentare.

Se invece si totalizzerà un punteggio esorbitante, non bisogna esaltarsi, perché scendere di grado è molto facile.

Supertank per C64

```

10 PRINT"[CLR]{HOME}":RESTORE:V=53248:POK
EV+32,0:POKEV+33,1:POKEV+17,PEEK(V+17)
AND247:rem 174
15 PRINTTAB(55)"[BLU]{RVS}BENVENUTO!":PRI
NTTAB(59)"[RED]{RVS}AL":rem 45
17 PRINTTAB(55)"[RED]{RVS}SUPERTANK!":PRI
NT:rem 158
20 FORQ=1TO4:PRINTTAB(14)"[<5>][<N>]:NEX
T:rem 50
22 PRINTTAB(14)"[<N>][RVS]~{ 8 SPAZI}
[<*>]:PRINTTAB(14)"[RVS]~ Q Q
{ 2 SPAZI}{ 3 Q} [<*>][OFF][< 2 I>]
[< 8 O>][< 3 I>":rem 209
24 PRINTTAB(13)"[RVS]~{ 2 SPAZI}Q Q
{ 3 SPAZI}M{ 3 SPAZI}[OFF][< 2 U>]
[< 8 Y>][< 3 U>":rem 207
26 PRINTTAB(13)"[RVS]{ 3 SPAZI}{ 3 Q}
{ 2 SPAZI}{ 3 Q}{ 2 SPAZI}[OFF]":
rem 79
28 PRINT"[ 3 SPAZI]{RVS}~{ 30 SPAZI}[<*>]
[OFF]":rem 103
30 PRINT"[ 2 SPAZI]{RVS}~{ 32 SPAZI}[<*>]
[OFF]":rem 96
32 PRINT"[GRN][< 2 +>][<5>][RVS]
{ 34 SPAZI}[OFF][GRN][< 3 +>":
rem 236
34 PRINT"[< 3 +>][<5>]M [RVS]~ [<*>]
[OFF]W[RVS]~ [<*>][OFF]W[RVS]~ [<*>]
[OFF]W[RVS]~ [<*>][OFF]W[RVS]~ [<*>]
[OFF]W[RVS]~ [<*>][OFF]W[RVS]~ [<*>]
[OFF]WN[GRN][< 4 +>":rem 255
36 PRINT"[< 4 +>][<5>]M [RVS] Q [OFF]
[RVS] Q [OFF] [RVS] Q [OFF] [RVS] Q
[OFF] [RVS] Q [OFF] [RVS] Q [OFF]
[RVS] Q [OFF]N[GRN][< 5 +>":rem 148
38 PRINT"[< 5 +>][<5>]M [<*>][RVS] [OFF]
~W [<*>][RVS] [OFF]~W [<*>][RVS] [OFF]
~W [<*>][RVS] [OFF]~W [<*>][RVS] [OFF]
~W [<*>][RVS] [OFF]~W [<*>][RVS] [OFF]
~[GRN][< 6 +>":rem 31
40 PRINT"[< 6 +>][<5>][< 26 Y>][GRN]
[< 7 +>":rem 239
42 FORQ=1TO2:PRINT"[< 39 +>]:NEXT
:rem 175
43 GOSUB800:rem 127
48 S=54272:PORL=STOS+24:POKEL,0:NEXT
:rem 14
50 PRINT"[CLR]{HOME}{WHT}":POKEV+32,1:POK
EV+33,0:PRINTTAB(120):rem 187
52 GOSUB1000:rem 168
60 PRINT"[CLR]{HOME}{WHT}":PRINTTAB(84)"C
ARATTERISTICHE DEI CARRI ARMATI"
:rem 225
62 PRINTTAB(49)"[RVS]{GRN}{ 2 SPAZI}ALTA
{ 2 SPAZI}[OFF]{ 3 SPAZI}[RVS]{YEL} ME
DIA [OFF]{ 2 SPAZI}[RVS]{WHT} BASSA
[OFF]":rem 103
64 PRINTTAB(40)"[GRN][RVS].TIGER[OFF]
{ 4 SPAZI}RES.{ 6 SPAZI}[YEL]FUOCO
{ 4 SPAZI}[WHT]VEL.":rem 175
66 PRINTTAB(40)"[YEL][RVS]2.T-34[OFF]
{ 5 SPAZI}[GRN]VEL.{ 6 SPAZI}[YEL]RES.
{ 5 SPAZI}[WHT]FUOCO":rem 189
68 PRINTTAB(40)"[WHT][RVS]3.SHERMAN[OFF]
{ 2 SPAZI}[GRN]FUOCO{ 5 SPAZI}[YEL]VEL
.{ 5 SPAZI}[WHT]RES.":PRINTTAB(120)
:rem 207
70 PRINT:INPUT"(HOME){ 15 GIU'}
{ 6 SPAZI}QUALE TANK DESIDERATE":TA
:rem 100

```

```

72 IFTA<1ORTA>3THEN80:rem 56
78 PRINT"[ 4 GIU']{ 8 SPAZI}PRONTO ALLA B
ATTAGLIA!":FORQ=1TO500STEP.5:NEXT:GOTO
85:rem 58
80 PRINT"[HOME]{ 15 GIU'}{ 9 SPAZI}NON E'
POSSIBILE!{ 4 SPAZI}":GOSUB1300:GOTO
70:rem 230
85 V=53248:GOSUB1100:rem 130
90 PRINT"[CLR]{HOME}":POKEV+32,0:POKEV+33
,1:rem 56
92 PRINTTAB(7)"[RVS][<5>]~ [<*>][OFF]
{ 4 SPAZI}[RVS]~ [<*>][OFF]{ 7 SPAZI}
[RVS]~ [<*>][OFF]{ 4 SPAZI}[RVS]~ [<*>]
[OFF]{ 4 SPAZI}[RVS][BLK]PUNTI"
:rem 21
94 PRINT"[ 2 SPAZI]{RVS}[<5>]~ [<*>][OFF]
{ 2 SPAZI}[RVS]~ { 2 SPAZI}[OFF]
{ 3 SPAZI}[RVS]~ { 2 SPAZI}[<*>][OFF]
{ 5 SPAZI}[RVS]~ { 2 SPAZI}[<*>][OFF]
{ 2 SPAZI}[RVS]~ { 2 SPAZI}[<*>][OFF]
{ 2 SPAZI}[BLK]~SC:rem 52
96 PRINT"[RVS][<5>]~ { 2 SPAZI}[<*>]~
{ 3 SPAZI}[<*>][OFF]{ 2 SPAZI}[RVS]
{ 4 SPAZI}[<*>][OFF]{ 2 SPAZI}[RVS]
~{ 5 SPAZI}[OFF] [RVS]~{ 4 SPAZI}[<*>]
[OFF]{ 2 SPAZI}[RVS][BLK]RES.:
rem 124
98 PRINT"[RVS][<5>]~ { 9 SPAZI}[<*>]~
{ 5 SPAZI}[<*>]~ { 6 SPAZI}~ { 6 SPAZI}
[<*>][OFF][BLK]~AR:rem 29
100 FORQ=1TO17:PRINT"[RVS]{GRN}[< 39 +>]
":NEXT:rem 64
110 PRINT"[RVS][<5>][<Q>]{ 9 C}[<W>][DES]
~{ 13 SPAZI}[<*>][DES][<Q>]{ 9 C}
[<W>":rem 65
112 PRINT"[RVS][<5>][<Q>]{ 9 C}[<W>]
{ 17 SPAZI}[<Q>]{ 9 C}[<W>":rem 129
114 GOSUB420:rem 172
120 CB=1:TI$="000000":rem 34
150 POKEV+21,15:POKE2040,13:POKEV+39,0:PO
KEV,170:POKEV+1,150:Y=170:X=150:SH=0
:rem 27
151 POKE2042,193:POKEV+41,11:RF=0:UT=110:
POKEV+42,0:POKE2043,195:rem 206
152 POKE2041,14:POKEV+40,0:POKEV+2,X1:POK
EV+3,Y1:rem 109
180 S=NOTPEEK(56321)AND15:U=SAND1:D=SAND2
:L=SAND4:R=SAND8:Y1=0:X1=0:rem 165
182 POKEV+23,0:POKEV+29,0:rem 189
185 IFUTHENX=X-M1:IFX<110THENX=X+M1
:rem 253
187 IFDTHENX=X+M1:IFX>180THENX=X-M1
:rem 247
189 IFRTHENY=Y+M1:IFY>245THENY=Y-M1
:rem 14
191 IFLTHENY=Y-M1:IFY<90THENY=Y+M1
:rem 205
200 POKEV,Y:POKEV+1,X:rem 59
210 J=NOTPEEK(56321)AND16:IFJ=16THENGOSUB
245:rem 189
230 BO=BO+.5:GOSUB310:GOTO180:rem 220
245 SH=SH+1:X1=X:MR=Y:HH=Y/2:GOSUB400
:rem 64
247 FORDD=DDTOHHSTEP5:POKEV+2,DD:POKEV+3,
X1:GOSUB330:NEXT:rem 224
250 POKE2041,15:FORDD=DDTOMRSTEP5:POKEV+2
,DD:POKEV+3,X1:GOSUB330:NEXT:rem 171
251 IF(PEEK(V+30)AND4)>0THENIF(PEEK(V+30)
AND4)>0THENGOSUB253:rem 110
252 X1=0:DD=0:Y1=0:POKEV+2,X1:POKEV+3,MR:
POKE2041,14:RETURN:rem 131

```

```

253 POKE2041,192:POKEV+23,2:POKEV+29,2
:rem 184
254 POKEV+3,X1-10:POKEV+2,DD-12:GOSUB410:
GOSUB495 :rem 204
258 FORRE=1TO500:NEXT:POKEV+23,0:POKEV+29
,0 :rem 152
260 X1=0:DD=0:Y1=0:POKEV+2,X1:POKEV+3,MR:
POKE2041,14:RETURN :rem 130
310 RF=RF+5:IFRF>215THEN350 :rem 95
315 POKEV+4,RF:POKEV+5,UT:RETURN:rem 79
330 RF=RF+1.8:IFRF>215THENRF=0 :rem 48
333 IFRF=0THEN340 :rem 243
335 POKEV+4,RF:POKEV+5,UT:RETURN:rem 81
340 RF=0:UT=110:RS=INT(RND(0)*60):UT=UT+R
S:GOTO330 :rem 61
350 F1=RF:F2=UT:POKE2042,194:POKEV+6,F1:P
OKEV+7,F2:GOSUB400 :rem 77
355 FORQ=F2TO150STEP.4:POKEV+7,Q:NEXT:POK
EV+29,8:POKEV+23,8:POKEV+6,F1-12
:rem 55
360 F2=150:FORQ=F2TO230STEP.6:POKEV+7,Q:N
EXT:GOSUB410 :rem 73
390 POKEV+6,0:POKEV+7,0:POKEV+4,0:POKEV+5
,0:RF=0:POKEV+23,0:POKEV+29,0
:rem 235
391 POKE2042,193:GOTO499 :rem 118
400 S=54272:FORL=STOS+24:POKEL,0:NEXT:POK
ES+5,9:POKES+6,16:POKES+24,15
:rem 70
405 POKES+4,129:POKES+1,34:POKES,75:RETUR
N :rem 91
410 S=54272:FORL=STOS+24:POKEL,0:NEXT:POK
ES+5,11:POKES+6,16:POKES+24,15
:rem 112
415 POKES+4,129:POKES+1,54:POKES,111:RETU
RN :rem 133
420 S=54272:FORL=STOS+24:POKEL,0:NEXT:POK
ES+5,11:POKES+6,56:POKES+24,15
:rem 117
425 POKES+4,129:POKES+1,51:POKES,97:RETUR
N :rem 96
495 SC=SC+10:CB=0:RN=INT(RND(0)*15):CB=CB
+RN:GOTO500 :rem 95
499 AR=AR-1 :rem 103
500 V=53248:PRINT"{HOME}":POKEV+32,0:POKE
V+33,CB :rem 248
501 RF=0:UT=110:RS=INT(RND(0)*60):UT=UT+R
S:RF=RF+1.5 :rem 211
502 PRINTTAB(7)"{RVS}[<5>]~[<*>]{OFF}
{ 4 SPAZI}{RVS}[<*>]{OFF}{ 7 SPAZI}
{RVS}[<*>]{OFF}{ 4 SPAZI}{RVS}[<*>]
{OFF}{ 4 SPAZI}{RVS}{BLK}PUNTI"
:rem 65
504 PRINT"{ 2 SPAZI}{RVS}[<5>]~[<*>]{OFF}
{ 2 SPAZI}{RVS}[<*>]{OFF}
{ 3 SPAZI}{RVS}[<*>]{OFF}
{ 5 SPAZI}{RVS}[<*>]{OFF}
{ 2 SPAZI}{RVS}[<*>]{OFF}
{ 2 SPAZI}{BLK}";SC :rem 96
506 PRINT "{RVS}[<5>]~[ 2 SPAZI][<*>]~
{ 3 SPAZI][<*>]{OFF}{ 2 SPAZI}{RVS}
{ 4 SPAZI][<*>]{OFF}{ 2 SPAZI}{RVS}
~{ 5 SPAZI}{OFF}{RVS}[< 4 SPAZI}
[<*>]{OFF}{ 2 SPAZI}{RVS}{BLK}RES.:"
:rem 168
508 PRINT"{RVS}[<5>]~[ 9 SPAZI][<*>]~
{ 5 SPAZI][<*>]~[ 6 SPAZI]~
{ 6 SPAZI][<*>]{OFF}{BLK}";AR
:rem 73
509 IFAR=0THEN549 :rem 254
510 RETURN :rem 118
549 LL=18:BL=12:BB=15 :rem 169

```

```

550 PRINT"{HOME}":POKEV+32,0:POKEV+33,1
:rem 214
558 FORQ=5TO7:PRINTTAB(Q)"{BLK}M"SPC(10)"
M":NEXT :rem 41
560 PRINTTAB(7)"{BLK}N"SPC(11)"M"SPC(4)"
NM" :rem 29
561 PRINTTAB(6)"N"SPC(13)"M"SPC(2)"N"SPC(
2)"M" :rem 58
562 PRINTTAB(5)"{BLK}N"SPC(15)"{BLK}MN"SP
C(4)"M" :rem 177
563 PRINTTAB(5)"M"SPC(21)"N"SPC(3)"NM"
:rem 142
564 PRINTTAB(6)"M"SPC(6)"NM"SPC(11)"N"SPC
(3)"N"SPC(2)"M" :rem 136
565 PRINTTAB(7)"M"SPC(4)"N"SPC(2)"M"SPC(1
0)"M"SPC(2)"N"SPC(4)"M" :rem 52
566 PRINTTAB(8)"M"SPC(2)"N"SPC(4)"M"SPC(1
0)"MN"SPC(6)"M" :rem 139
567 PRINTTAB(9)"MN"SPC(6)"M"SPC(17)"N"
:rem 158
568 PRINTTAB(18)"M"SPC(15)"N":FORLB=1TO6:
PRINTTAB(LL)"N"SPC(14)"N":LL=LL-1:NEX
T :rem 60
570 FORQ=1TO5:PRINTTAB(BL)"N"SPC(BB)"M":B
L=BL-1:BB=BB+2:NEXT :rem 187
580 RESTORE:POKEV+23,0:POKEV+29,0:POKEV+2
1,0:GOSUB420:FORQ=1TO500STEP.1:NEXT
:rem 66
585 S=54272:FORL=STOS+24:POKEL,0:NEXT:GOS
UB1200 :rem 193
588 V=53248:BO=BO/10:XX=INT(BO):SC=SC+XX:
IFSC>HSTHENHS=SC :rem 174
589 PRINT"{HOME}{CLR}":POKEV+32,0:POKEV+3
3,1:POKE53281,1 :rem 62
590 PRINTTAB(85)"{RVS}[<2>]B*O*N*U*S ";XX
;SPC(3)"{RVS}BOMBE USATE";SH:PRINTTAB
(45)"{RVS}TUOI"; :rem 167
591 PRINT "PUNTI";SC;SPC(3)"{RVS}* RECORD
* ";HS:PRINTTAB(49)"{RVS}GRADO =
{BLK}";BS :rem 66
592 PRINTTAB(43)"{RVS}[<2>]SEI SOPRAVISS
UTO PER: ";TIS :rem 189
593 PRINTTAB(90)"{RVS}GIOCHI ANCORA? (S/N
)" :rem 220
595 GETCS:IFCS=" "THEN595 :rem 109
596 SC=0:BS=" ":IFCS="S"THEN599 :rem 162
597 IFCS="N"THENEND :rem 112
598 CS=" ":GOTO595 :rem 164
599 SH=0:SC=0:BO=0:XX=0:POKEV+32,1:POKEV+
33,0:GOTO60 :rem 172
800 S=54272:FORL=STOS+24:POKEL,0:NEXT:POK
ES+5,9:POKES+6,0:POKES+24,15:rem 19
801 READHF,LF,DR:IFHF=-1THENRETURN :rem 196
804 POKES+1,HF:POKES,LF:POKES+4,33:FORT=1
TODR:NEXT:POKES+4,32:GOTO801:rem 168
810 DATA18,209,1024,15,210,512,18,209,512
,16,195,1024,14,24,512,11,48,512
:rem 45
811 DATA18,209,200,16,195,200,15,210,200,
14,24,200,15,210,512,22,96,512
:rem 178
812 DATA16,195,1024,11,48,1024,15,210,512
,14,24,200,12,143,200,11,48,200
:rem 222
813 DATA10,143,200,11,48,200,12,143,200,1
1,48,512,16,195,512,14,239,512
:rem 185
814 DATA11,48,512,15,210,200,15,210,200,1
4,24,200,12,143,200,11,48,200
:rem 105

```

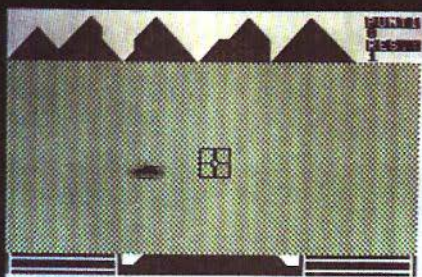
```

815 DATA10,143,200,11,48,200,12,143,200,1
    1,48,512,16,195,1024,22,96,512
    :rem 186
816 DATA18,209,1024,15,210,512,18,209,512
    16,195,1024,14,24,512,11,48,512
    :rem 51
817 DATA18,209,200,16,195,200,15,210,200,
    14,24,200,15,210,512,22,96,512
    :rem 184
818 DATA16,195,1024,11,48,1024,15,210,512
    14,24,200,12,143,200,11,48,200
    :rem 228
819 DATA10,143,200,11,48,200,12,143,200,1
    1,48,512,16,195,512,14,239,512
    :rem 191
820 DATA11,48,512,15,210,200,15,210,200,1
    4,24,200,12,143,200,11,48,200
    :rem 102
821 DATA10,143,200,11,48,200,12,143,200,1
    1,48,512,11,48,1024,-1,0,0 :rem 210
900 DATA255,255,255,128,24,1,128,24,1,128
    24,1,128,24,1,128,24,1,128,24,1
    :rem 35
905 DATA128,24,1,128,36,1,128,66,1,255,12
    9,255,128,66,1,128,36,1,128,24,1,128
    :rem 1
910 DATA24,1,128,24,1,128,24,1,128,24,1,1
    28,24,1,128,24,1,255,255,255:rem 93
915 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,0,0,31,254,0,49,255,192,96,255
    :rem 88
920 DATA240,196,127,252,206,127,255,206,1
    27,255,196,127,252,96,255,240,49,255
    :rem 35
925 DATA192,31,254,0,0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,0,0 :rem 109
926 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,0,0,63,240,0,103,252,0,195,255
    :rem 66
927 DATA0,219,255,192,195,255,0,103,252,0
    63,240,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    :rem 207
928 DATA0,0,0,0,0,0,0,0,0 :rem 113
930 DATA0,0,2,34,128,4,0,64,2,146,128,16,
    0,16,10,73,32,64,0,4,17,140,96,64,0,4
    :rem 2
935 DATA17,17,16,64,0,4,8,136,136,64,0,4,
    17,17,16,32,0,8,8,136,128,16,0,16,2,7
    2 :rem 80
940 DATA128,4,0,64,0,0,0 :rem 184
945 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,64,0,0,
    64,0,0,64,0,0,64,0,0,126,0,0,255,255,
    0 :rem 221
947 DATA255,0,63,255,252,127,255,254,255,
    255,255,100,68,70,37,85,84,20,68,72
    :rem 243
950 DATA15,255,240,0,0,0,0,0,0,0,0,0,0,0
    :rem 228
955 DATA0,2,0,0,2,0,0,2,0,0,2,0,0,58,0,0,
    70,0,0,186,0,0,130,0,0,254,0,1,255,0
    :rem 128
960 DATA3,255,128,7,255,192,0,124,0,15,18
    7,224,8,186,32,15,187,224,8,130,32,15
    :rem 52
965 DATA131,224,0,0,0,0,0,0,0,0,0,0,0,0
    :rem 83
970 DATA0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,60,0,0,126,0,0,255,0,0,255,0,0
    :rem 41
975 DATA126,0,0,60,0,0,0,0,0,0,0,0,0,0,0,0
    0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
    :rem 178

```

LOAD

Supertank
per C64



```

1000 FORA1=832TO894:READQ1:POKEA1,Q1:NEXT
    :rem 22
1010 FORA2=896TO958:READQ2:POKEA2,Q2:NEXT
    :rem 38
1015 FORA3=960TO1022:READQ3:POKEA3,Q3:NEX
    T :rem 70
1020 FORA4=12288TO12350:READQ4:POKEA4,Q4:
    NEXT :rem 226
1025 FORA5=12352TO12414:READQ5:POKEA5,Q5:
    NEXT :rem 228
1030 FORA6=12416TO12478:READQ6:POKEA6,Q6:
    NEXT :rem 239
1035 FORA7=12480TO12542:READQ7:POKEA7,Q7:
    NEXT :rem 241
1090 RETURN :rem 170
1100 IFTA=1THENAR=5:IFTA=1THENM1=1
    :rem 111
1105 IFTA=2THENAR=3:IFTA=2THENM1=2
    :rem 117
1110 IFTA=3THENAR=1:IFTA=3THENM1=3
    :rem 114
1150 RETURN :rem 167
1200 IFSC=0THENB$="BURBA" :rem 215
1201 IFSC>0ANDSC<51THENB$="SOLDATO"
    :rem 142
1202 IFSC>52ANDSC<101THENB$="CAPORALE"
    :rem 35
1204 IFSC>101ANDSC<201THENB$="SERGENTE"
    :rem 103
1206 IFSC>201ANDSC<401THENB$="TENENTE"
    :rem 34
1208 IFSC>401ANDSC<601THENB$="CAPITANO"
    :rem 100
1210 IFSC>601ANDSC<801THENB$="MAGGIORE"
    :rem 93
1212 IFSC>801ANDSC<1001THENB$="* COLONNEL
    LO *" :rem 136
1214 IFSC>1001THENB$="** GENERALE **"
    :rem 238
1216 RETURN :rem 170
1232 GOTO500 :rem 150
1300 FORI=1TO1500:NEXT:RETURN :rem 94

```

E' IN EDICOLA

IL MEGLIO DI **he**

SUPPLEMENTO AL N° 11 DELL'E L. 8.000

**IL MEGLIO DEL SOFTWARE
PER IL TUO COMPUTER IN CASA: GIOCHI,
ADVENTURE GAME, GESTIONE DELLA CASA
DIDATTICA, GRAFICA,
MUSICA, UTILITY**



PROVE E RECENSIONI

GRUPPO
EDITORIALE
JACKSON



**IN APPENDICE GUIDA
ALL'ACQUISTO DEL TUO
PERSONAL COMPUTER
SECONDO
LE TUE ESIGENZE**

**UN'OCCASIONE
DA NON PERDERE**



**GRUPPO
EDITORIALE
JACKSON**

San Francisco - Londra - Milano

VIC sound

A tutto Bach! per C64

di A. e M. Fossati

Questo mese proponiamo ai lettori di SUPERCOMMODORE un arrangiamento per C64 di un brano notissimo di musica classica: il secondo movimento (andante) del Concerto brandeburgese n. 2 di Johann Sebastian Bach, sicuri di fare cosa gradita ai moltissimi estimatori di questo "grande" della musica di tutti i tempi.

Bach scrisse i 6 Concerti Brandeburgesi nel 1721 durante il suo soggiorno a Kothen, forse il periodo più prolifico della sua vita. I Concerti furono scritti, come era costume comune al tempo, in dedica ad un principe, il marchese Cristiano Ludovico di Brandeburgo.

Per quanto riguarda il nostro programma esso è suddiviso in 2 sezioni: la prima, molto breve (linee 1-970), permette di inserire nel SID gli opportuni valori di durata e frequenza presenti nella seconda parte, cioè le successive istruzioni DATA (linee 1000-1652). Non ci resta che augurare buona digitazione e, soprattutto, buon ascolto di questa musica straordinaria, diretta e interpretata da un solista d'eccezione: il vostro C64.



A tutto Bach! per C64

```

1 REM ANDANTE                                     :rem 16
10 S=54272:W1=S+4:W2=W1+7:W3=W2+7:V1=S:V2=
   S+7:V3=S+14                                     :rem 100
50 PRINTCHR$(14):PRINTCHR$(8):POKE53281,1
   4:POKE646,0                                     :rem 92
55 PRINT"{CLR}{ 5 GIU' }A. & M. FOSSATI
   { 25 SPAZI }[< 15 T>]"                         :rem 89
60 PRINT"{ 5 GIU' }DAL CONCERTO BRANDEBURG
   HESE N. 2 "                                       :rem 233
65 PRINT"{GIU' }DI J. {SPAZI}S. BACH:"
   :rem 59
70 PRINT"{GIU' }ARRANGIAMENTO PER COMMODORE
   E 64 DEL SE-"                                     :rem 161
75 PRINT"{GIU' }CONDO MOVIMENTO (ANDANTE)."
   :rem 215
100 O1=017:POKES+05,052:POKES+06,085
   :rem 11
110 O2=017:POKES+12,085:POKES+13,169
   :rem 18

```

```

120 O3=033:POKES+19,009:POKES+20,085
   :rem 16
130 POKES+24,15                                     :rem 57
200 READQ:ONQOTO300,350,400,950,960,970,
   450,500,550,600,900                             :rem 171
250 READP:FORF=0:TOPSTEP.062:NEXT:rem 69
290 GOTO200                                         :rem 102
300 POKEW1,0:READLB,HB:POKEV1,LB:POKEV1+1
   ,HB:POKEW1,O1                                     :rem 137
340 GOTO200                                         :rem 98
350 POKEW2,0:READLB,HB:POKEV2,LB:POKEV2+1
   ,HB:POKEW2,O2                                     :rem 147
390 GOTO200                                         :rem 103
400 POKEW3,0:READLB,HB:POKEV3,LB:POKEV3+1
   ,HB:POKEW3,O3                                     :rem 148
440 GOTO200                                         :rem 99
450 POKEW1,0:GOTO200                               :rem 177
500 POKEW2,0:GOTO200                               :rem 174
550 GOSUB590:POKEV1+1,HB+2                         :rem 23
555 GOSUB590:POKEV1+1,HB                           :rem 191
560 GOSUB590:POKEV1+1,HB+2                         :rem 24

```

```

565 GOSUB590:POKEV1+1,HB:GOTO200:rem 197
590 FORR=1TO80:NEXT:RETURN :rem 227
600 GOSUB590:POKEV2+1,HB+2 :rem 20
605 GOSUB590:POKEV2+1,HB :rem 188
610 GOSUB590:POKEV2+1,HB+2 :rem 21
615 GOSUB590:POKEV2+1,HB:GOTO200:rem 194
900 FORV=15TO0STEP-.02:POKES+24,V:NEXT:EN
D :rem 158
950 READLB,HB:POKEV1,LB:POKEV1+1,HB:POKEW
1,01:GOTO200 :rem 76
960 READLB,HB:POKEV2,LB:POKEV2+1,HB:POKEW
2,02:GOTO200 :rem 81
970 READLB,HB:POKEV3,LB:POKEV3+1,HB:POKEW
3,03:GOTO200 :rem 86
1000 DATA7,8,3,001,000,0,90 :rem 63
1011 DATA3,196,009,0,30 :rem 140
1012 DATA3,162,014,0,30 :rem 130
1013 DATA3,158,011,0,30 :rem 133
1014 DATA3,162,014,0,30 :rem 132
1015 DATA3,196,009,0,30 :rem 144
1016 DATA3,162,014,0,30 :rem 134
1021 DATA3,010,013,0,30 :rem 121
1022 DATA3,129,015,0,30 :rem 135
1023 DATA3,247,010,0,30 :rem 132
1024 DATA3,010,013,0,30 :rem 124
1025 DATA3,056,009,0,30 :rem 140
1026 DATA3,247,010,0,30 :rem 135
1031 DATA3,196,009,0,30 :rem 142
1032 DATA3,162,014,0,30 :rem 132
1033 DATA3,158,011,0,30 :rem 135
1034 DATA3,162,014,0,30 :rem 134
1035 DATA3,196,009,2,069,029,0,29 :rem 138
:rem 138
1041 DATA3,010,013,2,003,031,0,29 :rem 96
:rem 133
1042 DATA3,247,010,0,30 :rem 140
1043 DATA3,056,009,0,30 :rem 140
1044 DATA3,247,010,5,069,029,0,29 :rem 130
:rem 138
1051 DATA3,196,009,2,059,023,0,05,10 :rem 8
:rem 136
1052 DATA3,081,007,0,30 :rem 116
1053 DATA3,207,005,2,237,021,0,29 :rem 128
:rem 104
1055 DATA3,226,004,2,118,046,1,138,058,0, :rem 138
28 :rem 88
1056 DATA3,207,005,0,30 :rem 229
1061 DATA3,133,006,2,118,046,1,005,062,0, :rem 133
28 :rem 126
1062 DATA3,193,007,5,219,043,0,29,8 :rem 46
:rem 65
1063 DATA3,123,005,0,30 :rem 203
1064 DATA3,133,006,1,138,058,0,29 :rem 226
:rem 123
1065 DATA3,081,007,2,219,043,1,039,052,0, :rem 127
12,4,005,062,0,13 :rem 65
1066 DATA3,162,014,4,138,058,5,118,046,0, :rem 203
12,5,039,052,0,13 :rem 226
1071 DATA3,196,009,2,219,043,1,118,046,0, :rem 123
05,9 :rem 127
1072 DATA3,162,014,5,018,039,0,29,8 :rem 127
:rem 127
1073 DATA3,158,011,1,219,043,0,29 :rem 127
:rem 127
1074 DATA3,162,014,4,018,039,0,29 :rem 127
:rem 127
1075 DATA3,196,009,2,059,023,1,138,058,0, :rem 116
28 :rem 142
1076 DATA3,158,011,0,30 :rem 97
1081 DATA3,232,006,2,059,023,1,181,065,0, :rem 225
28 :rem 135
1082 DATA3,055,008,5,237,021,0,29,8 :rem 135
:rem 92
1083 DATA3,123,005,0,30 :rem 65
1084 DATA3,232,006,4,157,069,0,13,4,036,0 :rem 148
78,0,14 :rem 195
1085 DATA3,081,007,2,059,023,1,157,069,0, :rem 225
12,4,181,065,0,13 :rem 130
1086 DATA3,196,009,4,157,069,0,29 :rem 130
:rem 130
1091 DATA3,247,010,2,059,023,1,065,055,0, :rem 130
05,9 :rem 110
1092 DATA3,109,016,5,237,021,0,29,8 :rem 140
:rem 106
1093 DATA3,208,013,1,058,049,0,29 :rem 223
:rem 134
1094 DATA3,109,016,4,219,043,0,29 :rem 124
:rem 48
1095 DATA3,162,014,2,219,043,1,157,069,0, :rem 69
28 :rem 91
1096 DATA3,103,017,0,30 :rem 203
1101 DATA3,137,019,2,118,046,1,157,069,0, :rem 119
28 :rem 116
1102 DATA3,059,023,4,181,065,0,29,7 :rem 235
:rem 130
1103 DATA3,109,016,0,30 :rem 102
1104 DATA3,137,019,2,219,043,0,29 :rem 227
:rem 133
1105 DATA3,010,013,1,181,065,2,019,039,0, :rem 119
12,5,138,058,0,13 :rem 50
1106 DATA3,109,016,5,039,052,4,157,069,0, :rem 122
12,4,036,078,0,13 :rem 92
1111 DATA3,103,017,2,118,046,1,181,065,0, :rem 53
28 :rem 136
1112 DATA3,237,021,5,219,043,4,138,058,0, :rem 135
28,7 :rem 114
1113 DATA3,162,014,2,018,039,0,29 :rem 141
:rem 100
1114 DATA3,103,017,5,207,034,0,29 :rem 100
:rem 133
1115 DATA3,158,011,1,138,058,2,219,032,0, :rem 133
04,10 :rem 133
1116 DATA3,162,014,5,069,029,0,29 :rem 119
:rem 119
1121 DATA3,109,016,2,018,039,1,138,058,0, :rem 122
28 :rem 92
1122 DATA3,137,019,4,065,055,0,29,7 :rem 53
:rem 136
1123 DATA3,208,013,0,30 :rem 135
1124 DATA3,109,016,2,207,034,0,29 :rem 114
:rem 141
1125 DATA3,247,010,1,065,055,2,219,032,0, :rem 100
12,5,018,039,0,13 :rem 100
1126 DATA3,237,021,5,207,034,0,29 :rem 100
:rem 100
1131 DATA3,162,014,2,219,032,1,065,055,0, :rem 100
28 :rem 100
1132 DATA3,103,017,5,069,029,4,138,058,0, :rem 100
28,7,8 :rem 100
1133 DATA3,158,011,0,30 :rem 100
1134 DATA3,162,014,0,30 :rem 100
1135 DATA3,196,009,2,118,046,1,138,058,0, :rem 100
28 :rem 100
1136 DATA3,055,008,0,30 :rem 100
1141 DATA3,247,010,2,118,046,1,138,058,0, :rem 100
28 :rem 100
1142 DATA3,196,009,5,219,043,0,29 :rem 100
:rem 100

```

1143 DATA3,247,010,2,018,039,0,29 :rem 122
 1144 DATA1,181,065,5,207,034,0,29 :rem 126
 1145 DATA3,123,005,2,018,039,1,219,043,0,13,5,118,046,4,138,058,0,13:rem 36
 1146 DATA5,219,043,4,065,055,0,29 :rem 134
 1151 DATA3,081,007,2,018,039,1,138,058,0,28 :rem 104
 1152 DATA3,247,010,5,207,034,0,29,7 :rem 219
 1153 DATA3,180,008,2,219,032,0,05,10 :rem 1
 1154 DATA3,247,010,5,069,029,0,29 :rem 132
 1155 DATA3,081,007,2,207,034,1,157,069,0,28 :rem 106
 1156 DATA3,180,008,0,30 :rem 142
 1161 DATA3,193,007,2,018,039,1,157,069,0,28 :rem 112
 1162 DATA3,196,009,4,005,062,0,29,7 :rem 227
 1163 DATA3,133,006,0,30 :rem 136
 1164 DATA3,193,007,5,219,043,0,14,5,118,046,0,14 :rem 88
 1165 DATA3,090,004,2,219,043,1,005,062,0,12,5,039,052,0,13 :rem 45
 1166 DATA3,180,008,5,118,046,4,157,069,0,12,4,036,078,0,13 :rem 76
 1171 DATA3,207,005,2,118,046,1,005,062,0,28 :rem 91
 1172 DATA3,081,007,4,138,058,0,29,7 :rem 231
 1173 DATA3,133,006,2,219,043,0,05,10 :rem 1
 1174 DATA3,193,007,5,018,039,0,29 :rem 135
 1175 DATA3,081,007,2,207,034,1,138,058,0,28 :rem 105
 1176 DATA3,180,008,5,018,039,0,29 :rem 134
 1181 DATA3,193,007,2,207,034,1,138,058,0,28 :rem 106
 1182 DATA3,196,009,5,003,031,4,039,052,0,28,7,8 :rem 46
 1183 DATA3,133,006,0,30 :rem 138
 1184 DATA3,193,007,0,30 :rem 146
 1185 DATA3,180,008,2,003,031,1,039,052,0,28 :rem 92
 1186 DATA3,103,017,5,207,034,4,138,058,0,13,5,018,039,4,005,062,0,13:rem 38
 1191 DATA3,158,011,2,003,031,1,039,052,0,28 :rem 88
 1192 DATA3,162,014,5,069,029,4,118,046,0,28,7,8 :rem 56
 1193 DATA3,010,013,0,30 :rem 131
 1194 DATA3,129,015,0,30 :rem 145
 1195 DATA3,162,014,2,069,029,1,157,069,0,28 :rem 118
 1196 DATA3,103,017,0,30 :rem 141
 1201 DATA3,129,015,2,069,029,1,036,078,0,28 :rem 106
 1202 DATA3,137,019,5,020,026,0,29,8 :rem 217
 1203 DATA3,010,013,0,30 :rem 123
 1204 DATA3,129,015,4,182,087,0,14,4,237,092,0,14 :rem 89
 1205 DATA3,056,009,2,020,026,1,182,087,0,12,4,078,104,0,13 :rem 51
 1206 DATA3,247,010,4,237,092,5,069,029,0,12,5,003,031,0,13 :rem 53

1211 DATA3,196,009,1,237,092,2,020,026,0,05,10 :rem 233
 1212 DATA3,162,014,5,059,023,0,29,8 :rem 220
 1213 DATA3,158,011,1,182,087,0,05,9 :rem 221
 1214 DATA3,162,014,4,036,078,0,29 :rem 126
 1215 DATA3,196,009,2,018,039,1,157,069,0,28 :rem 117
 1216 DATA3,158,011,4,181,065,0,29 :rem 127
 1221 DATA3,247,010,2,018,039,1,078,104,0,28 :rem 95
 1222 DATA3,158,011,5,207,034,0,29 :rem 120
 1223 DATA3,010,013,2,207,034,1,237,092,0,28 :rem 86
 1224 DATA5,219,032,4,182,087,0,29 :rem 134
 1225 DATA3,133,006,2,219,032,1,036,078,0,28 :rem 98
 1226 DATA4,157,069,5,069,029,0,14,5,219,032,0,14 :rem 100
 1231 DATA3,180,008,2,207,034,1,157,069,0,28 :rem 102
 1232 DATA3,247,010,0,30,7,8 :rem 77
 1233 DATA3,196,009,0,30 :rem 146
 1234 DATA3,158,011,0,30 :rem 138
 1235 DATA3,247,010,2,039,052,0,29 :rem 122
 1236 DATA3,010,013,0,30 :rem 129
 1241 DATA3,158,011,2,138,058,0,29 :rem 127
 1242 DATA3,196,009,0,30 :rem 146
 1243 DATA3,055,008,0,30 :rem 140
 1244 DATA3,196,009,2,039,052,0,29 :rem 133
 1245 DATA3,133,006,2,118,046,0,14,5,138,058,0,14 :rem 84
 1246 DATA3,055,008,5,039,052,0,29 :rem 131
 1251 DATA3,180,008,2,219,043,0,05,10 :rem 2
 1252 DATA3,081,007,5,039,052,0,14,5,058,049,0,14 :rem 88
 1253 DATA3,193,007,2,039,052,0,29 :rem 128
 1254 DATA3,039,006,5,005,062,0,14,5,138,058,0,14 :rem 85
 1255 DATA3,133,006,2,005,062,1,036,078,0,28 :rem 97
 1256 DATA3,193,007,0,30 :rem 146
 1261 DATA3,180,008,2,005,062,1,201,082,0,28 :rem 87
 1262 DATA3,089,010,5,138,058,0,29,8 :rem 235
 1263 DATA3,081,007,0,30 :rem 140
 1264 DATA3,180,008,1,036,078,0,29 :rem 131
 1265 DATA3,196,009,2,138,058,1,157,069,0,13,4,201,082,0,14 :rem 74
 1266 DATA3,226,004,4,036,078,5,005,062,0,13,5,157,069,0,14 :rem 70
 1271 DATA1,157,069,3,133,006,4,005,062,2,138,058,0,05,10 :rem 220
 1272 DATA3,247,010,2,039,052,1,036,078,0,13,4,193,073,0,14,8 :rem 161
 1273 DATA3,158,011,1,036,078,0,29 :rem 130
 1274 DATA3,056,009,1,237,092,0,14,4,182,087,0,14 :rem 95



1275 DATA3,196,009,2,069,029,1,237,092,0,28 :rem 123
 1276 DATA3,158,011,0,30 :rem 144
 1281 DATA3,010,013,2,003,031,1,237,092,0,28 :rem 81
 1282 DATA3,129,015,4,182,087,0,29,7 :rem 236
 1283 DATA3,247,010,0,30 :rem 140
 1284 DATA3,010,013,2,069,029,0,29 :rem 124
 1285 DATA3,162,014,2,020,026,1,182,087,0,12,5,003,031,0,13 :rem 41
 1286 DATA3,081,007,5,069,029,4,237,092,0,12,4,078,104,0,13 :rem 75
 1291 DATA3,196,009,2,059,023,1,182,087,0,05,9 :rem 213
 1292 DATA3,162,014,4,036,078,0,29,7 :rem 231
 1293 DATA3,158,011,2,237,021,0,29 :rem 124
 1294 DATA3,162,014,5,137,019,0,29 :rem 132
 1295 DATA3,196,009,2,118,046,0,29 :rem 140
 1296 DATA3,158,011,0,30 :rem 146
 1301 DATA3,010,013,2,118,046,0,29 :rem 108
 1302 DATA3,129,015,5,101,041,0,29,8 :rem 212
 1303 DATA3,089,010,0,30 :rem 137
 1304 DATA3,010,013,0,30 :rem 125
 1305 DATA3,180,008,2,101,041,0,29 :rem 111
 1306 DATA3,089,010,5,118,046,0,14,5,039,0,52,0,14 :rem 84
 1311 DATA3,158,011,2,101,041,0,05,10 :rem 242
 1312 DATA3,162,014,5,018,039,0,29,8 :rem 223
 1313 DATA3,196,009,0,30 :rem 145
 1314 DATA3,158,011,0,30 :rem 137
 1315 DATA3,193,007,2,018,039,1,237,092,0,28 :rem 108
 1316 DATA3,196,009,0,30 :rem 148

1321 DATA3,089,010,2,018,039,1,078,104,0,28 :rem 100
 1322 DATA3,180,008,5,207,034,0,29,8 :rem 222
 1323 DATA3,158,011,0,30 :rem 137
 1324 DATA1,237,092,0,30 :rem 143
 1325 DATA3,207,005,2,207,034,1,201,082,0,13,4,078,104,0,14 :rem 43
 1326 DATA4,237,092,5,018,039,0,13,5,101,0,41,0,14 :rem 80
 1331 DATA3,193,007,2,207,034,1,201,082,0,28 :rem 91
 1332 DATA3,158,011,5,003,031,4,036,078,0,28,7 :rem 195
 1333 DATA3,196,009,2,118,046,0,29 :rem 133
 1334 DATA3,158,011,0,30 :rem 139
 1335 DATA3,193,007,2,005,062,1,018,039,0,28 :rem 100
 1336 DATA3,196,009,0,30 :rem 150
 1341 DATA3,207,005,2,005,062,1,018,039,0,28 :rem 91
 1342 DATA3,103,017,5,138,058,4,207,034,0,28,8,7 :rem 46
 1343 DATA3,162,014,0,30 :rem 137
 1344 DATA3,103,017,0,30 :rem 136
 1345 DATA3,158,011,2,138,058,0,29 :rem 132
 1346 DATA3,162,014,0,30 :rem 140
 1351 DATA3,180,008,2,138,058,0,29 :rem 130
 1352 DATA3,010,013,5,039,052,0,29,8 :rem 216
 1353 DATA3,089,010,1,039,052,0,29 :rem 126
 1354 DATA3,010,013,0,30 :rem 130
 1355 DATA3,180,008,1,157,069,0,29 :rem 136
 1356 DATA3,089,010,0,30 :rem 145
 1361 DATA3,133,006,1,157,069,0,29 :rem 129
 1362 DATA3,137,019,4,005,062,0,29,7 :rem 225
 1363 DATA3,129,015,2,018,039,0,29 :rem 130
 1364 DATA3,137,019,0,30 :rem 147
 1365 DATA3,010,013,2,039,052,1,005,062,0,28 :rem 86
 1366 DATA3,129,015,0,30 :rem 146
 1371 DATA3,196,009,2,039,052,1,005,062,0,28 :rem 103
 1372 DATA3,162,014,5,058,049,4,138,058,0,28,8,7 :rem 61
 1373 DATA3,078,012,0,30 :rem 144
 1374 DATA3,162,014,0,30 :rem 141
 1375 DATA3,010,013,2,005,062,1,036,078,0,28 :rem 92
 1376 DATA3,129,015,0,30 :rem 147
 1381 DATA3,103,017,2,005,062,1,201,082,0,28 :rem 85
 1382 DATA3,178,020,5,138,058,0,29,8 :rem 238
 1383 DATA3,162,014,0,30 :rem 141
 1384 DATA3,103,017,1,036,078,0,29 :rem 129
 1385 DATA3,158,011,2,138,058,1,157,069,0,12,4,078,104,0,13 :rem 73
 1386 DATA3,162,014,4,237,092,5,005,062,0,12,5,157,069,0,13 :rem 70
 1391 DATA3,129,015,1,201,082,2,138,058,0,04,10 :rem 239

1392 DATA3,137,019,5,039,052,4,036,078,0,
 28,8 :rem 219
 1393 DATA3,010,013,1,157,069,0,05,9
 :rem 221
 1394 DATA3,129,015,4,005,062,0,29
 :rem 128
 1395 DATA3,089,010,2,028,039,1,138,058,0,
 28 :rem 117
 1396 DATA3,010,013,4,039,052,0,29
 :rem 123
 1401 DATA3,162,014,2,101,041,1,157,069,0,
 28 :rem 89
 1402 DATA3,103,017,0,30 :rem 131
 1403 DATA3,078,012,0,30 :rem 138
 1404 DATA3,162,014,2,018,039,1,005,062,0,
 28 :rem 91
 1405 DATA3,196,009,2,207,034,1,138,058,0,
 12,5,101,041,4,157,069,0,12:rem 38
 1406 DATA3,137,019,5,018,039,4,005,062,0,
 28 :rem 106
 1411 DATA3,010,013,2,207,034,1,138,058,0,
 05,9 :rem 183
 1412 DATA3,129,015,5,003,031,4,039,052,0,
 28,7 :rem 191
 1413 DATA3,089,010,2,069,029,0,29
 :rem 131
 1414 DATA3,010,013,5,020,026,0,29
 :rem 106
 1415 DATA3,180,008,2,069,029,1,202,082,0,
 28 :rem 102
 1416 DATA3,081,007,5,003,031,0,14,5,207,0
 34,0,14 :rem 68
 1421 DATA3,196,009,1,201,082,2,003,031,0,
 05,10 :rem 223
 1422 DATA3,180,008,4,036,078,0,29
 :rem 130
 1423 DATA3,196,009,1,157,069,0,29
 :rem 140
 1424 DATA2,069,029,4,005,062,0,29
 :rem 129
 1425 DATA3,226,004,1,157,069,2,069,029,0,
 13,4,201,082,0,13 :rem 61
 1426 DATA5,003,031,4,036,078,0,29
 :rem 126
 1431 DATA3,133,006,2,020,026,1,157,069,0,
 28 :rem 94
 1432 DATA3,137,019,4,005,062,0,29,8
 :rem 224
 1433 DATA3,129,015,1,138,058,0,29
 :rem 131
 1434 DATA3,137,019,4,039,052,0,29,7
 :rem 231
 1435 DATA3,010,013,2,018,039,0,29
 :rem 117
 1436 DATA3,129,015,0,30 :rem 144
 1441 DATA3,056,009,2,219,043,0,29
 :rem 127
 1442 DATA3,247,010,0,30 :rem 137
 1443 DATA3,081,007,0,30 :rem 140
 1444 DATA3,056,009,5,118,046,0,14,5,039,0
 52,0,14 :rem 89
 1445 DATA3,196,009,1,003,031,2,118,046,0,
 12,5,219,043,0,13 :rem 53
 1446 DATA3,133,006,5,118,046,0,29
 :rem 129
 1451 DATA3,081,007,1,003,031,2,225,036,0,
 05,10 :rem 222
 1452 DATA3,247,010,4,069,029,0,29,7
 :rem 231
 1453 DATA3,056,009,2,219,032,0,29
 :rem 128
 1454 DATA3,247,010,5,069,029,0,29,8
 :rem 235

1455 DATA3,081,007,1,039,052,0,29
 :rem 127
 1456 DATA3,056,009,0,30 :rem 148
 1461 DATA3,226,004,1,039,052,0,29
 :rem 122
 1462 DATA3,162,014,4,118,046,0,29,7
 :rem 226
 1463 DATA3,158,011,0,30 :rem 142
 1464 DATA3,162,014,0,30 :rem 141
 1465 DATA3,196,009,2,018,039,0,29
 :rem 140
 1466 DATA3,158,011,0,30 :rem 145
 1471 DATA3,081,007,2,028,039,0,29
 :rem 129
 1472 DATA3,247,010,5,225,036,0,29,8
 :rem 227
 1473 DATA3,056,009,0,30 :rem 147
 1474 DATA3,247,010,0,30 :rem 142
 1475 DATA3,081,007,2,219,043,1,020,026,0,
 28 :rem 96
 1476 DATA3,056,009,0,30 :rem 150
 1481 DATA3,226,004,2,219,043,1,020,026,0,
 28 :rem 91
 1482 DATA3,162,014,5,018,039,4,059,023,0,
 28,8,7 :rem 52
 1483 DATA3,158,011,0,30 :rem 144
 1484 DATA3,162,014,0,30 :rem 143
 1485 DATA3,196,009,1,118,046,0,29
 :rem 140
 1486 DATA3,158,011,0,30 :rem 147
 1491 DATA3,081,007,1,118,046,0,29
 :rem 128
 1492 DATA3,247,010,4,219,043,0,29,7
 :rem 228
 1493 DATA3,056,009,2,237,021,0,29
 :rem 130
 1494 DATA3,247,010,0,30 :rem 144
 1495 DATA3,081,007,2,020,026,0,29
 :rem 123
 1496 DATA3,056,009,0,30 :rem 152
 1501 DATA3,226,004,2,020,026,0,29
 :rem 109
 1502 DATA3,162,014,5,157,024,0,29,8
 :rem 222
 1503 DATA3,078,012,1,138,058,0,29
 :rem 129
 1504 DATA3,162,014,0,30 :rem 136
 1505 DATA3,196,009,2,058,049,1,157,069,0,
 28 :rem 124
 1506 DATA3,078,012,0,30 :rem 142
 1511 DATA3,133,006,2,058,049,1,157,069,0,
 28 :rem 109
 1512 DATA3,137,019,5,039,052,4,181,065,0,
 28,8,7 :rem 53
 1513 DATA3,109,016,0,30 :rem 139
 1514 DATA3,137,019,0,30 :rem 144
 1515 DATA3,010,013,1,181,065,0,29
 :rem 115
 1516 DATA3,109,016,0,30 :rem 142
 1521 DATA3,180,008,1,181,065,0,29
 :rem 124
 1522 DATA3,010,013,4,157,069,0,29,7
 :rem 222
 1523 DATA3,247,010,2,020,026,0,29
 :rem 113
 1524 DATA3,010,013,0,30 :rem 129
 1525 DATA3,180,008,2,003,031,0,29
 :rem 115
 1526 DATA3,247,010,0,30 :rem 140
 1531 DATA3,207,005,2,003,031,0,29
 :rem 109
 1532 DATA3,103,017,5,069,029,0,29,8
 :rem 230

1533 DATA3,162,014,1,207,034,0,29 :rem 119
 1534 DATA3,103,017,0,30 :rem 137
 1535 DATA3,158,011,2,069,029,1,101,041,0,28 :rem 97
 1536 DATA3,162,014,0,30 :rem 141
 1541 DATA3,193,007,2,069,029,1,101,041,0,28 :rem 98
 1542 DATA3,158,011,5,003,031,4,018,039,0,28,8,7 :rem 39
 1543 DATA3,196,009,0,30 :rem 150
 1544 DATA3,158,011,0,30 :rem 142
 1545 DATA3,193,007,1,118,046,0,29 :rem 132
 1546 DATA3,196,009,0,30 :rem 153
 1551 DATA3,123,005,1,118,046,0,29 :rem 120
 1552 DATA3,129,015,4,219,043,0,29,7 :rem 229
 1553 DATA3,010,013,2,005,062,0,29 :rem 110
 1554 DATA3,129,015,0,30 :rem 145
 1555 DATA3,247,010,2,018,039,0,29 :rem 129
 1556 DATA3,010,013,0,30 :rem 134
 1561 DATA3,081,007,2,018,039,0,29 :rem 128
 1562 DATA3,247,010,5,225,036,0,29,8 :rem 227
 1563 DATA3,056,009,1,219,043,0,29 :rem 131
 1564 DATA3,247,010,0,30 :rem 142
 1565 DATA3,081,007,2,219,043,1,039,052,0,28 :rem 105
 1566 DATA3,056,009,0,30 :rem 150
 1571 DATA3,226,004,2,219,043,1,039,052,0,28 :rem 100
 1572 DATA3,162,014,5,018,039,4,118,046,0,28,8,7 :rem 53
 1573 DATA3,158,011,0,30 :rem 144
 1574 DATA3,162,014,0,30 :rem 143
 1575 DATA3,196,009,1,118,046,0,29 :rem 140
 1576 DATA3,158,011,0,30 :rem 147
 1581 DATA3,010,013,1,118,046,0,29 :rem 117
 1582 DATA3,129,015,4,219,043,0,29,7 :rem 232
 1583 DATA3,247,010,0,30 :rem 143
 1584 DATA3,010,013,0,30 :rem 135
 1585 DATA3,180,008,1,219,043,0,29 :rem 132
 1586 DATA3,247,010,4,118,046,0,14,4,039,0,52,0,14 :rem 88
 1591 DATA3,158,011,1,219,043,0,29 :rem 128
 1592 DATA3,162,014,4,018,039,0,29,7 :rem 231
 1593 DATA3,196,009,0,30 :rem 155
 1594 DATA3,158,011,0,30 :rem 147
 1595 DATA3,193,007,2,036,078,1,018,039,0,28 :rem 119
 1596 DATA3,196,009,0,30 :rem 158
 1601 DATA3,247,010,2,005,062,1,018,039,0,28 :rem 90
 1602 DATA3,010,013,4,225,036,0,29,7 :rem 211
 1603 DATA3,056,009,0,30 :rem 142
 1604 DATA3,247,010,2,138,058,0,29 :rem 128
 1605 DATA3,081,007,1,225,036,2,039,052,0,12,5,005,062,0,13 :rem 46

1606 DATA3,162,014,5,138,058,0,29 :rem 133
 1611 DATA3,196,009,2,039,052,1,225,036,0,28 :rem 105
 1612 DATA3,158,011,5,118,046,4,018,039,0,28,7 :rem 206
 1613 DATA3,193,007,2,219,043,0,29 :rem 128
 1614 DATA3,196,009,5,018,039,0,29 :rem 139
 1615 DATA3,133,006,2,219,043,1,018,039,0,28 :rem 100
 1616 DATA3,123,005,5,118,046,0,14,5,039,0,52,0,14 :rem 81
 1621 DATA3,081,007,1,018,039,2,118,046,0,05,10 :rem 237
 1622 DATA3,133,006,0,30 :rem 136
 1623 DATA3,081,007,0,30 :rem 140
 1624 DATA2,219,043,1,219,043,0,29 :rem 127
 1625 DATA2,219,043,1,069,029,0,14,4,018,0,39,0,14 :rem 91
 1626 DATA5,118,046,4,225,036,0,29 :rem 135
 1631 DATA3,193,007,2,225,036,1,039,052,0,28 :rem 102
 1632 DATA0,31 :rem 166
 1633 DATA4,039,052,0,15,4,005,062,0,15 :rem 103
 1634 DATA2,018,039,4,138,058,0,29 :rem 139
 1635 DATA1,039,052,3,055,008,2,160,027,4,118,046,0,27 :rem 81
 1636 DATA0,31 :rem 170
 1641 DATA6,055,008,5,020,026,4,118,046,0,13,4,225,036,0,14 :rem 56
 1642 DATA2,069,029,4,018,039,0,29 :rem 139
 1643 DATA3,081,007,1,018,039,2,059,023,0,05,10 :rem 240
 1644 DATA0,31 :rem 169
 1645 DATA2,237,021,4,225,036,0,05,9 :rem 223
 1646 DATA5,137,019,0,15,1,018,039,0,14 :rem 114
 1651 DATA3,226,004,2,137,019,1,018,039,0,28 :rem 103
 1652 DATA0,60,11 :rem 56



Gli ultimi saranno i primi?

Giudiciale tu.

Anche se, in ordine di tempo, siamo gli ultimi a uscire con libri "dedicati" ai più diffusi home e personal computer, siamo convinti che il lettore attento ed esigente apprezzerà la qualità dei contenuti e la loro presentazione.

Abbiamo selezionato e tradotto i best-seller mondiali che accolgono quanto di più utile è stato scritto per il tuo computer.

Scegli a colpo sicuro!

Puoi ordinare direttamente i titoli presentati compilando ed inviando il coupon pubblicato oppure acquistarli presso i più qualificati computer shop e le migliori librerie.

CEDOLA DI ORDINAZIONE - LIBRI

Compilare e spedire in busta chiusa a:
J. soft - Viale Restelli 5 - 20124 Milano
tel. 02/6880841-6880842-6880843

Ordino i seguenti libri per un importo totale di L.
+ L. 2.000 come contributo fisso per spese di spedizione

Cod. Cod.
Cod. Cod.
Cod. Cod.

- ☐ Contanti allegati
☐ Assegno allegato n°
☐ Ho spedito l'importo a mezzo vaglia postale
☐ Ho versato l'importo sul CCP n° 19445204 intestato a J. soft - Milano
☐ Pagherò in contrassegno al postino al ricevimento dei volumi

Nome

Cognome

Via

CAP Città Prov.

Se richiesta fattura - codice fiscale

Data

J. soft: libri di qualità

Per Commodore 64

Giochi fantastici per il C 64
Cod. ASOC 001 L. 42.000
(inclusa cassetta)

Un libro pensato e realizzato per il divertimento dell'utente del C 64; contiene 19 giochi di vario genere tutti ampiamente commentati. L'attento studio dei programmi potrà essere vanfagioso per l'apprendimento delle tecniche di programmazione.

Il libro del C 64
Volume 1
Cod. ASOC 010 L. 24.000
(inclusa cassetta)
Il libro del C 64
Volume 2
Cod. ASOC 011 L. 24.000
(inclusa cassetta)

I due libri illustrano a fondo le possibilità del Commodore 64. Un compendio di utili consigli, e quant'altro è necessario per conoscere meglio il proprio calcolatore. Esempi pratici completano l'esposizione in modo chiaro ed esauriente.



Inoltre, i Paper Book: la raccolta dei programmi suddivisi per macchina di Paper Soft, il primo settimanale di software su carta per il tuo computer.

Paper Book - programmi per Apple II -
Cod. ASOC 005 L. 18.000
Paper Book - programmi per VIC 20 -
Cod. ASOC 003 L. 18.000
Paper Book - programmi per Commodore 64 -
Cod. ASOC 004 L. 18.000
Paper Book - programmi per Texas TI 99/4A -
Cod. ASOC 007 L. 18.000
Paper Book - programmi per ZX Spectrum -
Cod. ASOC 006 L. 18.000

J. soft s.r.l.

Viale Restelli, 5 - 20124 Milano - tel. 02/6880841-6880842-6880843



C 64: suono e grafica - Volume 1
Cod. ASOC 008 L. 24.000
(inclusa cassetta)
C 64: suono e grafica - Volume 2
Cod. ASOC 009 L. 24.000
(inclusa cassetta)

Due volumi che non possono mancare nella biblioteca dell'utente Commodore. Una miniera di idee e suggerimenti per la programmazione del tuo personal computer. Una vasta serie di programmi esemplificativi guidano il lettore al miglior sfruttamento del Commodore 64.

Per ZX Spectrum

15 giochi grafici per ZX Spectrum
Cod. ASOC 002 L. 24.000
(inclusa cassetta)

Un fantastico volume con 15 entusiasmanti giochi grafici per il tuo ZX Spectrum. Tutti i programmi sono "pronti

all'uso" e ampiamente commentati. I programmi sono scritti sfruttando appieno le capacità del calcolatore ed il loro studio può giovare a chiunque intenda affinare le proprie tecniche di programmazione.



In ogni volume, corredato di cassetta, oltre 40 programmi per il tuo computer: giochi grafici, utility, arcade, adventure una miscela esplosiva di software "pronto all'uso"!

P R I N T

di M. Cristuib Grizzi

Recensioni Libri

COME USARE IL TUO COMMODORE 64: linguaggio macchina, ingressi/ uscite e unità periferiche



di **F. Monteil**
ed. **Il Rostro**

Il C64 è diffuso nel nostro paese ormai dalla metà del 1983, e molti sono i possessori di questo computer che hanno sviluppato una buona conoscenza della macchina e della sua programmazione in BASIC. Una volta acquisita questa esperienza nell'uso in BASIC del C64, nasce spontaneamente in quasi tutti i programmatori il desiderio - a metà tra curiosità e necessità - di imparare il linguaggio macchina per ottenere dal computer tutte quelle funzioni altrimenti impossibili se programmate in BASIC. Inoltre, chi possiede unità periferiche, quali stampanti e disk drive, sente spesso il desiderio di ottenere prestazioni

sempre migliori, e ciò è possibile solo con una buona conoscenza sia hardware che software delle unità interessate.

Questo libro si propone di guidare il lettore - che ovviamente si suppone già esperto della programmazione in BASIC - nel nuovo mondo del linguaggio macchina: dopo il primo capitolo, dedicato all'esposizione della struttura hardware del C64 ed in particolare alla sua mappa di memoria, si imparerà come memorizzare delle routine in linguaggio macchina entro un programma BASIC, come accedere alle routine (già pronte!) del sistema operativo e come realizzare, tramite il I/m, grafica in alta risoluzione ed animazioni. Un altro capitolo affronta i problemi degli ingressi e delle uscite, argomento di estremo interesse per chi intendesse interfacciare il computer ad apparati casalinghi, come il termostato del riscaldamento, lo scaldabagno, ecc.

Infine un ultimo capitolo è dedicato alle periferiche ed ai programmi già pronti, disponibili per il loro sfruttamento ottimale.

Data la notoria difficoltà che in genere si riscontra nell'approccio al linguaggio macchina ed al sistema di numerazione binario, ben vengano volumi come questo che possono chiarire subito i concetti più "ostici" ed aprire velocemente la strada ad una diretta utilizzazione di quanto imparato.

COMMODORE 64 LA GRAFICA E IL SUONO

di **R. Bonelli, L. Pazzucconi,
F. Racchi, G. Valerio**
ed. **Jackson**

Seguito logico di altri due volumi dedicati rispettivamente al BASIC ed alla gestione file del C64 - già precedentemente recensiti su queste pagine - que-

st'ultimo tratta gli argomenti forse più affascinanti in un personal che si rispetti: la grafica e gli effetti sonori.

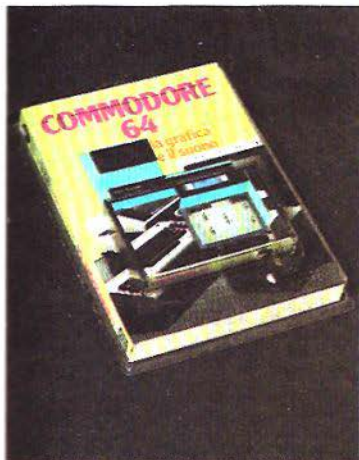
Il libro è strutturato su quattro distinti capitoli, partendo dalle applicazioni grafiche elementari ottenibili direttamente da tastiera e man mano approfondendo l'analisi di tutti i procedimenti per ottenere immagini più complesse. Particolare spazio ed attenzione vengono dedicati all'analisi della programmazione che utilizza la pagina in alta risoluzione del C64 e gli otto SPRITE disponibili; con l'uso di questi ultimi vengono proposti interessanti programmi di animazione, che permettono di realizzare emozionanti videogiochi. I concetti esposti a livello teorico vengono immediatamente applicati in programmi esemplificativi listati nel volume e contenuti già registrati in una cassetta allegata allo stesso, risultando quindi di facile ed immediato apprendimento anche per il principiante. Tra questi programmi ne spiccano alcuni, per il tracciamento di superfici in alta risoluzione, in grado di realizzare grafici veramente spettacolari (sul tipo di quelli realizzati nella sigla della trasmissione "Quark"...).

Un intero capitolo viene dedicato al suono, argomento diverso, ma complementare alla grafica: esso tratta tutte le possibili applicazioni dei tre generatori indipendenti implementati sul C64, da quelle più semplici, ma necessarie per comprenderne il funzionamento, fino a programmi avanzati, che comprendono l'uso dei filtri, modulazione ad anello, sincronizzazione e variazione dinamica del suono.

In appendice è compreso un elenco delle locazioni di memoria preposte al controllo del VIC-II (il chip che presiede alla gestione del video nel C64) e del SID (il chip che presiede invece alla gestione del suono), unitamente ad una tabella con le frequenze delle varie note musicali.

Tutti i listati di programma compresi nel

volume sono stampati con cura e risultano pertanto di buona leggibilità: la cassetta allegata offre inoltre il vantaggio di poter osservare sul video il funzionamento di un programma, analizzandone contemporaneamente il listato sulle pagine del libro.



I libri vengono offerti con il 10% di sconto sul prezzo di copertina

I libri recensiti possono essere ordinati utilizzando il coupon a pag. 82



Dalla grande edicola Jackson

Tutto sull'hobby e home computer

he

HOME COMPUTER

La rivista del computer in casa. "Prove su strada" di software e programmi per tutti i personal computer.

11 numeri all'anno: L. 3.500 a numero

Abbonamento: solo L. 31.500

**VIDEO
Giochi**

La guida indiscussa al fantastico mondo dei videogames. La più eccitante, divertente, istruttiva rassegna del settore.

11 numeri all'anno: L. 3.500 a numero

Abbonamento: solo L. 31.000

elektor

Il mensile di elettronica venduto in mezzo milione di copie e redatto in 7 lingue. Con articoli su: applicazioni, progettazioni, sperimentazioni, invenzioni.

11 numeri all'anno: L. 3.000 a numero

Abbonamento: solo L. 29.000

**strumenti
MUSICALI**

Il mondo delle 7 note in versione...elettronica. Con test strumentali, novità e analisi del mercato, servizi speciali.

10 numeri all'anno: L. 3.000 a numero

Abbonamento: solo L. 24.000

Quando l'informazione fa testo

In busta chiusa inviate questo coupon a:

Gruppo Editoriale Jackson via Rosellini, 12 - 20124 Mi

☐ Desidero ricevere GRATIS un numero della Rivista

(allego L. 1.000 in francobolli per contributo spese di spedizione)

☐ Inviatemi GRATIS il Catalogo della Biblioteca JACKSON (allego L. 1.000 in francobolli per contributo spese di spedizione)

Nome _____

Cognome _____

via _____

CAP _____

Città _____

input-output

Piccoli annunci

Cambio software per C64 su disco e cassetta, anche ultime novità come: Summer Games, Impossible Mission, The Castle Of Doctor Greep, Turbo Disco Gyrruss, Disk Mimic e molti, molti altri. Offerta sempre valida. Massimiliano Bruno - Via C. Cittadini, 62 - 53100 Siena - Tel. 0577/54075

Vendo VIC 20 più registratore C2N, molto software, 8 Kbyte RAM, 3 Kbyte super expander, programma di Archivio, 3 libri. Il tutto, usato pochissimo e in imballo originale, a L. 600.000. Telefonare ore pasti. Paolo Zao - Via Del Castagnone, 33 - 15048 Valenza (AL) - Tel. 0131/951096

Causa passaggio a sistema superiore vendo VIC 20 più 6 programmi in cartuccia, 3 libri di programmi e utilizzo, imballo originale. Tutto in ottime condizioni a L. 240.000 trattabili. Telefonare dalle 19 in poi. Giancarlo Lacchin - Via A. Kramer, 6 - 20129 Milano - Tel. 02/793523

Vendo programmi per VIC 20 sia versione base che con espansione memoria anche blocco 5. Telefonare o scrivere per accordi. Gianni Gaudino - Via Grogia, 18 - 10136 Torino - Tel. 011/352830

Per CBM 64 vendo cassetta contenente 8 fantastici giochi come BC e One On One e il programma I miei amici al favoloso prezzo di L. 30.000. Per ricevere la cassetta telefonare a: Loris Zucchetti - Corso Milano, 4 - 25056 Ponte di Legno (BS) - Tel. 0364/91018

Vendo VIC 20 più registratore C2N a L. 200.000; inoltre vendo console Intellevision più 7 cartucce molto belle a L. 400.000 (solo console L. 230.000). Telefonare dopo le 14. Mario Barbero - Via Tiboldo Bolla, 25 - 13060 Tavigliano (VC) - Tel. 015/472134

Vendo VIC 20 più registratore C2N, 8 Kbyte RAM, super expander, 6 cassette con giochi ed utility, vari libri. Tutto in perfette condizioni a L. 350.000. Umberto Crotti - Via Lidice, 4 - 42015 Correggio (RE) - Tel. 0522/694144

Siamo due ragazzi di Gorizia che hanno il VIC 20 e lo Spectrum, vendiamo a prezzi bassissimi programmi di qualsiasi genere (anche copiatori). Chianese Flavio - Via Virgilio, 17 - 34170 Gorizia - Tel. 0481/33183

Fatti regalare un VIC 20 più avrai centinaia di programmi gratis. Il prezzo? Pochissimo: solo L. 100.000! Contattatemi, io sono Alessio Saltarin - Via Mailbran, 1 - 20090 Trezzano S/N (MI) - Tel. 4453193

Vendo per C64 centinaia di programmi a prezzi favolosi, da L. 5.000 a L. 15.000. The Last One L. 15.000, Flight II L. 10.000. Per cassette 6 programmi L. 20.000 inclusa cassetta. A richiesta anche manuali. N.B. Le spese del disco e di spedizione sono escluse. Luigi Palumbo - Viale Gramsci - 80122 Napoli - Tel. 081/66006

Vendo programmi gestionali per Commodore 64. Cassetta giochi per CBM 64, con ultime novità, L. 25.000. Programmi per VIC 20 a L. 500 cadauno (acquisto minimo 10 programmi). Manuali in italiano. Telefonare ore 20.30-21.30. Roberto - 20145 Milano - Tel. 02/389590

Vendo oltre 500 programmi per CBM 64 a prezzi stracciati. Ubaldo Mancini - Via Valdelattorre, 174 - 10151 Torino - Tel. 011/7391285

Vendo VIC 20 più registratore, 4 cartucce (fra cui "Gorf"), corso completo "Introduzione al BASIC" 1 e 2 parte (4 cassette più 2 libri) a L. 350.000 trattabili. Regalo listati e altri programmi. Antonino Famulari - Via Casina, 21 - 22038 Tavernerio (CO) - Tel. 031/420528

Cerco espansione per VIC 20, cambio con autoradio mangianastri stereo Philips 682 OM-FM oppure Sharp RG5900E ASTS (Automatic Stereo Tuning System) ANSS (Automatic Noise Suppressor System) FM - LW - MW. Telefonare dalle 13.30 alle 15. Renato Carramaschi - Via Rizza, 22 - 42010 Villarotta (RE) - Tel. 0552/828111

Vendo favolosi giochi per Commodore 64: Basket, Baseball, International Soccer, Turbo Tape, ecc. Per ricevere la lista inviare L. 1.000 in francobolli. Annuncio sempre valido. Andrea Monacelli - Via Carpianto - 06023 Camperi (PG)

Vendo, causa doppio regalo, VIC 20 ancora imballato più: registratore, espansione 16Kbyte, alimentatore, convertitore, 20 cassette gioco, 5 cartucce gioco, a sole L. 300.000. Marco Pendino - Via Ugo Betti, 25 - 20100 Milano - Tel. 02/3087174

Vendo programmi per VIC 20: potete scegliere tra 75 giochi (di cui 50 in LM) a prezzi eccezionali, ad esempio: cassetta C90 con 25 giochi L. 10.000, tutti in blocco L. 25.000; oppure studi da L. 500 a L. 1.500 cadauno. Richiedere lista, accludendo francobollo. Disponibili anche alcuni listati. Galanti Marcello - Via Torrenova, 134 - 00133 Roma

Vendo per VIC 20 cartuccia "The Cpunt" al prezzo di L. 10.000. Inoltre cambio giochi su cassetta. Scrivetemi! Federico Fioretti - Via Aurelia, 607/E - 00165 Roma - Tel. 06/6220963

Compro Commodore 64 più registratore; pago al massimo L. 600.000. Marcello Galanti - Via Torrenova, 134 - 00133 Roma

Cerco programmi di qualsiasi tipo su cassetta per C64. Mandare elenco a: Masello Alessandro - Via 4 Novembre - 20050 Ronco Briantino - Tel. 039/671287

Compro copia dei numeri 5 e 6 di "Commodore Computer Club". Franco Spada - Via Fardella, 444 - 91100 Trapani - Tel. 0923/22301

Vendo giochi listati e su cassetta per VIC 20. Sergio Marchese - Via G. Ricciardi, 12/B - 66013 Chieti Scalo - Tel. 0871/54612

Per VIC 20 dispongo di 100 programmi che vendo a prezzi eccezionali. Ricco e dettagliato catalogo inviando L. 1.000 a: Paolo Valchera - Via Del Poggio, 8 - 03100 Frosinone - Tel. 81159

Vendo come nuovo VIC 20, joystick, 2 cassette software (Star Battle, Road Race) a L. 280.000. Inoltre vendo oltre 150 programmi per VIC 20 espanso ed inespanso da L. 1.000 a 10.000. Marco Luna - Via Lorenzo il Magnifico, 20 - 00162 Roma - Tel. 06/4271839

Vendo, cambio giochi ed utility per Commodore 64. Sia su nastro che su disco. Richiedete la lista. Rispondo a tutti. Annuncio sempre valido. Alberto Martiniucci - Viale Elba, 61 - 57037 Portoferraio (LI) - Tel. 0565/92756 (ore pasti)

Vendo 100 giochi per CBM 64. Prezzo eccezionale! Solo L. 50.000! Giochi come Calcio, Pole Position, Xaxxon, Tarzan, Sam, Skramble, Basket, Forbidden Forest, Pitfall, Burger Time... Richiesta lista gratis; telefonare ore pasti e serali oppure scrivete a: Andrea Rota Nodari - Via Valosa di Sopra, 26 - 20052 Monza (MI) - Tel. 039/741597

Cambio, vendo software per CBM 64. Invio cassette o dischi con i miei programmi a chi invia le sue. Sono interessato particolarmente ad un programma di autorun per disco e cassette. Scrivetemi per scambio. Cerco anche libretto istruzioni 1541 in italiano. Vincenzo Bono - Via Umberto, 174 - 91021 Campobello di Mazara (TP)

Vendo eccezionali giochi per VIC 20 inespanso su cassetta: 8 giochi L. 16.000 soltanto. Posseggo anche giochi da 8 Kbyte che vendo a sole L. 8.000 l'uno, sempre su cassetta. Richiedi la lista. Giampaolo Cellini - Via Paradiso, 41 - 66100 Chieti - Tel. 61142 (ore pasti)

Vendo centinaia di programmi per C64: stupendi giochi, gestionali, utility, linguaggi. Alcuni esempi: Logo, FORTH, Pascal, Comal Austro Compiler, Pet Speed, Super Base, Calc Resul, Easy File, Easy Script, Magic Desk. Claudio Segatori - Via R. Fancelli, 23 - 00169 Roma - Tel. 06/2679183

Vendo, cambio 2 cartucce per C64, Avenger e Kick-Man contro Music Machine e Music Composer. Claudio Porcellana - Via Olmi, 5 - 10048 Vinovo (TO) - Tel. 011/9651559

Per VIC 20 cambio o invio contro rimborso spese programmi su cassetta. Assicuro risposta a tutti. Alvaro Caccarini - Via Di Vittorio, 10 - 58022 Follonica (GR)

Compro per C64 programmi giochi, utility, animazione, grafici, ecc., possibilmente su disco. Preferibile materiale di prim'ordine. Inviare liste molto dettagliate e prezzi. Preferisco trattare in blocco grosse quantità di programmi. Tita Gerosa - Via Monte Santo, 6 - 22053 Lecco (CO)

Cambio programmi e giochi per C64. Inviare liste, rispondo a tutti. Cedo 10 a scelta (su nastro) a L. 15.000. Telefonare ore 19.21.30. Gianni Cottogni - Via strambino, 23 - 10010 Carrone (TO)

Vendo per i possessori del CBM 64 bellissimi giochi, molti col Turbo Tape. Ne possiedo quasi 160 fra cui: Calcio, Pole Position, Frogger, Qbert, Defender, Fort Apocalypse, Hunch Bach. Prezzi bassissimi e da concordare. Belia Francesco - Via G. Leopardi, 91 - 95100 Catania - Tel. 095/378514

Cambio software per C64, oltre 500 programmi. Mandate le vostre liste a: Virginio Caretoni - Via Diaz, 5 - 20080 Cislano (MI)

Vendo eccezionali giochi, anche in 3D, per C64, in blocco o separatamente. Scrivete o telefonate. Inoltre vendo VIC 20 più registratore C2N, cassetta con 50 giochi. Riccardo Belardini - Via Prov. S. Montano, 18 - 80076 Lacco Ameno (NA) - Tel. 081/995453

Vendo cartridge "Raid On Fort Knox", per VIC 20 inespanso, a L. 45.000 trattabili. Scrivere o telefonare a: Maurizio Valle - Via D. Petri, 20 - 02100 Rieti - Tel. 496638

Vendo nuovissimi VIC 20 più registratore C2N, gioco "VIC Avenger" (cartridge), "Il libro del Commodore VIC 20", cassetta "Test Q. Intellettuale" e "Air Attack" a L. 300.000. Gian Mario Sini - Via Luna e Sole, 23 - 07100 Sassari - Tel. 079/290073

Cambio software per C64 su disco o cassetta. Possiedo numerosi programmi di gioco ed utility. Inviare le vostre liste ed io invierò le mie. Annuncio sempre valido. Massimiliano Sani - Via S. Rosa - 56048 Volterra (PT) - Tel. 88907

Per C64 vendo bellissimi programmi di utility - didattica per studenti dalle medie all'università su argomenti come fisica, matematica, geometria, chimica, ecc. Richiedete lista. Gabriele Raiti - Via C. Rosselli, 113 - 96016 Lentini (SR) - Tel. 945079

Vendo, cambio giochi per CBM 64 in LM tra cui: Hunck Back, Pole Position, Mr. Robot, Miner 2049'er, Dig Dug, Robotron, Comer's Tower, One On One, Pit Stop, James Bond 007, Slinky, Air Patrol, Forest. Alessandro Del Ferro - Via Paterni, 77 - 61100 Pesaro - Tel. 0721/67156

input·output

Piccoli
annunci

Per C64 cambio, compro, vendo software di qualsiasi genere su disco. Dispongo di tutto. Livio Modena - Via Dante, 110 - 18038 Sanremo (IM) - Tel. 018/473079

Vendo cassetta con 15 videogiochi per CBM 64 a L. 30.000; contiene fra gli altri: International Soccer, Radar Rat Race, Quix, ecc. Scambio programmi, scrivere o telefonare ore pasti. Massimo Marini - Via S. Ermete, 377 - 56100 Pisa - Tel. 050/982656

Se spedite L. 1.000 potrete ricevere la mia lista di programmi per VIC 20 e C64, listati o su cassetta. Vendo inoltre, per C64, "Dichiarazioni e Fatture IVA" a L. 6.000. Giovanni D'Ambrosio - Piazza S. Erasmo, 16 - 80142 Napoli (NA) - Tel. 081/281654

Compro, vendo, cambio software per VIC 20. Certi giochi sono rari e divertenti. Affrettatevi! Cerco disperatamente: "Wacky Waiters" e "Arcadia". Vendo: "La Principessa e lo Stregone", ecc. James Newman - Vicolo Copernico, 5 - 20020 Ceriano Laghetto Saronno (MI) - Tel. 02/9660300

Vendo per C64 cassette contenenti i seguenti giochi in LM: Unga pazio 2000, Trappola, Togo, Attack alcon Patrol, Super Pipeline, Bilardo, Burger Time, per L. 3.000 a gioco. In omaggio Turbotape più a scelta Briscola o Totocalcio. Telefonare ore pasti. Raffaele Iaccarino - Via S. Antonio - 88064 Chiaravalle Calabrese (CZ) - Tel. 0967/91665

Vendo VIC 20 più espansione 16 Kbyte, 1 cartidge, 15 cassette di giochi, 6 manuali sul VIC e su giochi a L. 300.000. Andrea Mastromatteo - Via Pesola, 33 - 71100 Foggia - Tel. 0881/27760

Compro programmi di vario genere per C64 (giochi, utility, ecc.) con Turbo Tape. Inviare liste a: Giuliano Presutto - Via Brignano Inf. 6463 - 84100 Salerno

Vendo stampante plotter Commodore 1520 come nuova a L. 300.000. Telefonare ore pasti. Nicola Salerno - Via Aspromonte, 8 - 37126 Verona - Tel. 04548692

Vendo per VIC 20 a prezzi stracciati software favoloso, su giochi (in LM) che utilities. G. Battista Gallus - Via S. Paolo, 3 - 09047 Selargius (CA) - Tel. 070/842795

Vendo Console Atari VCS 2600, solo tre mesi di vita, più 7 giochi tipo Mouse Trap, E.T., Pallavolo, Demon Attack, ecc. Il tutto completo di garanzia, a sole L. 600.000 trattabili. Telefonare ore pasti. Flavio Baldi - Via Dei Nacci, 1 - 53045 Montepulciano (SI) - Tel. 757926

Cerco colleghi sessantatristri per scambiare o vendere programmi. Ne possiedo di magnifici, tutti in LM. Vi aspetto! Scrivete o telefonate ore pasti. Antonio Vitale - Via Michele Titone, 61 - 90129 Palermo - Tel. 426273

Vendo V-Visitors, Sette 1/2, Dig Dag, Labirinto tridimensionale e un simpatico "VIC piano" a sole L. 6.500. Vendo a sole L. 27.500 la cartidge "The Count", avventure sul tuo VIC. Fabio Germanà - Via Minissale, 3 - 98013 Messina (Contesse) - Tel. 090/21964

Vendo per VIC 20 i migliori programmi su cassetta (Tornado, Arcadia, Poker, Cyclon Zap, ecc.). Invio il catalogo dei giochi a chi lo richiede (allegando L. 500). Alberto Frabetti - Via Ricciarelli, 139 - 44100 Ferrara - Tel. 0532/61034

Vendo Extended BASIC per CBM 64 con istruzioni in italiano a L. 40.000. Pietro Niddu - Via Bianchi, 1 - 07100 Sassari

Cambio software per CBM 64 nella sola zona di Catania. Annuncio sempre valido. Fabio Bellasai - Via F.lli Bandiera, 13 - 95100 Catania - Tel. 095/515353

Vendo VIC 20 4 mesi di vita più cartidge e programmi su nastro. Prezzo eccezionale, solo L. 100.000. Telefonare ore pasti. Fabio Ivaldi - Via Valle Migliorini, 4 - 10025 Pino Torinese (TO) - Tel. 840765

Cambio programmi per C64. Mandatemi le vostre liste di programmi ed io vi spedirò le mie. Ciao! Scrivete a: Francesco Paganelli - Via Rimini, 2 - 47038 Sant'Arcangelo di Romagna (FO)

Vendo 10 programmi per CBM 64 a L. 25.000, spese di spedizione comprese, tra cui: Burger Time, Grand Master, Fort Apocalypse, King Kong, Cudoly Cubert, ecc. Scrivere a Francesco Paganelli - Via Rimini, 2 - 47038 Sant'Arcangelo di Romagna (FO)

Vendo per CBM 64 programmi (gestionali - videogames - utilities - ecc.). Richiedere ampia lista descrittiva a: Lauro Michelotti - Via Boboli - 51017 Pienza (PT)

Vendo VIC 20 in ottime condizioni, completo di garanzia per 9 mesi, con: manuale in italiano, joystick, 80 programmi (Frog, Pac Man, Crazy Kong, alcune utility, ecc.); in imballo originale a L. 300.000 trattabili. Carlo Sechi - Via Colonia Romana, 4 - 07046 Porto Torres (SS) - Tel. 501394

Attenzione: vendo, per comprare libri, VIC 20 più espansione 32 Kbyte selezionabile, 20 riviste, 73 videogiochi su nastro, 4 giochi su cartidge. Il tutto a sole L. 500.000. Telefonate dalle 17.00 in poi. Vi prego, aiutatem! Giorgio Andreolotti - Via Andrea Doria, 17 - 21013 Gallarate (VA) - Tel. 797740

Vendo VIC 20, registratore, joystick, cassette "Multi Tron" e "Milipede", cartuccia "Cosmic Cruncher" a L. 400.000. In più regalo cassette con giochi, 9 lezioni, un package in LM. Tutto in ottimo stato. Telefonate ore pasti. Fabio Mangione - Via Teocrito, 48 - 95131 Catania - Tel. 095/322463

Per VIC 20 dispongo di una vasta nastroteca comprendente programmi di videogiochi, grafica, didattica, gestionali, giochi di società, per la casa, ecc., che vendo a prezzi da sballo. Ricco e dettagliato elenco inviando L. 1.000 (anche in francoboli), oppure elenco più cassetta dimostrativa inviando L. 10.000 a: Walter Vanzetti - Residence Faul Via Ivrea, 74 - 10016 Montalto Dora (TO)

Per Commodore 64 acquisto software su cassetta, Algebra, utility, ecc., gestionali. Inviare lista con caratteristiche e prezzi. Massima serietà. Spedire a: Rosario Orsanto - Via Cupa Vincinale Vallone, 20 - 80145 Milano (NA) - Tel. 081/7553937

Per C64 vendo, cambio software in LM (games, utility) tutti originali. Prezzi buoni. Scrivere o telefonare a: Cristiana Garibaldi - Vicolo S. Lazzaro - 18038 Sanremo (IM) - Tel. 82354

Vendo per CBM 64 programmi come Zaxxon, Skirace, Pitstop, ecc. Sono interessato a scambiare un gioco per il Totocalcio. Giulio Marazzina - Via IV Novembre, 36 - 20070 Massalengo (MI) - Tel. 02/78950

Serve un programma o una procedura realizzata su Commodore 64 o Olivetti M20, da un programmatore professionista? Allora, solo per la Lombardia: Silvestro Gabba - Via Riccardo Rossi, 54/1 - 27020 Domo, (PV) - Tel. 0382/848085

Vendo VIC 20 più registratore, 2 cartucce Commodore, manuale in italiano, libro "Alla scoperta del VIC 20", 2 cassette piene di programmi a sole L. 335.000. Alessandro Marras - Via A. Vivaldi, 8 - 40050 Montevoglio (BO) - Tel. 051/837448

Vendo per VIC 20 40 videogames allo strepitoso prezzo di L. 5.000; per lista giochi telefonare alle ore 20 o scrivere. Offerta sempre valida. Massimo Gagliardi - Viale Cannadoni, 70 - 62100 Macerata - Tel. 0733/30901

Vendo per VIC 20 4 cartucce: Gorf, Omega Race, Jupiter Lander, Snake Byte a L. 25.000 l'uno più L. 2.000 per spese di spedizione. Fabio Ferrari - Via Araldi, 5 - 29100 Piacenza - Tel. 0523/753283

Vendo moltissimi programmi di giochi, utility e didattica per CBM 64, su disco o listati, a buon prezzo. Massima serietà. Annuncio sempre valido. Telefonare ore pasti o scrivere per richiedere la lista. Olderico Caviglia - Via D. Carbone, 4 - 15060 Villafranca (AL) - Tel. 0131/8315

Vendo VIC 20 più registratore, joystick, manuale "Alla scoperta del VIC 20", 10 cassette di videogiochi, cavetti, libro istruzioni. Tutto usato pochissimo e in perfetto stato a sole L. 200.000. Giuseppe Zitarosa - Via Cuniolo, 12 - 15057 Tortona (AL) - Tel. 0131/812342

Causa passaggio a CBM 64, vendo VIC 20 più 70 giochi su cassetta a sole L. 200.000 (non trattabili). Il sistema è in ottimo stato ed efficientissimo. Pregasi telefonare ore pasti. Luca Monticelli - Via F.lli Bressan, 4 - 20156 Milano - Tel. 02/2551762

Vendo ottimi programmi in LM e BASIC a prezzi molto convenienti: a L. 3.500 - 4.500 ciascuno. Vendo inoltre cartuccia Visible Solar System a L. 15.000. Cerco espansione 8-16 Kbyte. Massimiliano Attona - Via Vittorio Veneto, 2 - 33089 Villota di Chions (PN) - Tel. 0434/630170

Vendo, cambio giochi, utility e word processor. Dispongo di 300 titoli tra i quali: Zaxxon, Jumpman, Flight Simulator, Donkey Kong. Antonino Molteni - Via Borgognone, 8 - 22063 Cantù (CO)

Vendo espansione 16 Kbyte RAM per VIC 20 a L. 100.000 non trattabili. Telefonare ore pasti: Mario Lombardi - Via Palmanova, 209 - 20132 Milano - Tel. 2567039

Vendo VIC 20, relativi cavetti, espansione 8 Kbyte, espansione 3 Kbyte super expander e tantissime cassette a sole L. 300.000: una vera offerta, tutto in ottime condizioni. Telefonare solo se veramente interessati. Giampaolo Cellini - Via Paradiso, 41 - 66100 Chieti - Tel. 0871/61142

Vendo software per VIC 20. Videogames e utility anche in LM. Magnifici videogames per VIC espanso (8-16 Kbyte). Richiedere lista a: Stefano Ronconi - Via A. Costa, 2 - 20026 Novate Milanese (MI) - Tel. 35471443

Vendo in blocco VIC 20 più trasformatore, cavi collegamento TV, registratore dedicato Commodore, joystick Pointmaster, istruzioni originali, 90 giochi su cassetta, 3 giochi su cartidge e decine di listati. Tutto perfettamente funzionante a L. 400.000. Vero affare. Stefano Zibardi - Via Piana, 28 - 20155 Milano - Tel. 02/3270318

Clamoroso! Vendo per VIC 20 in espanso una cassetta contenente 5 giochi da scegliere tra i seguenti: Pac Man, Crazy Kong, Defender, Meteors, Lama, Gridrunner, ecc. Il tutto a sole L. 15.000. Umberto Benelli - Via XX Settembre, 180 - 54031 Carrara-Avenza (MS) - Tel. 0585/57145

Vendo e cambio programmi per VIC 20. I prezzi sono bassissimi. L'offerta è sempre valida. Aspetto lettere o telefonate. Vi prego di darmi sempre il vostro telefono. Marco Zandonadi - Via G. Galilei, 11 - 20052 Monza (MI) - Tel. 039/835920

SUPERCOMMODORE

CECOLA DI ORDINAZIONE - LIBRI
da compilare e spedire in busta chiusa a:
J.Leaff - Viale Realelli, 5 - 20124 Milano - Tel. 688228

LIBRI

COME USARE IL TUO COMMODORE 64
vol. 2
COMMODORE 64
la grafica e il suono
cod. AROK003 L. 11.700
cod. AJAC045 L. 30.600

Ordino i seguenti libri per un importo totale di L. 2.000
come contributo fisso per spese di spedizione

Cod.Cod.Cod.

Cod.Cod.Cod.

☐ Contanti allegati ☐ Assegno allegato n°

☐ Ho spedito l'importo a mezzo vaglia postale

☐ Ho versato l'importo sul cc. n° 19445204 intestato a J.Leaff - Milano

☐ Pagherò in contrassegno al postino al ricevimento dei volumi (valido solo per l'Italia)

Nome
Cognome
Via
CittàC.A.P.Prov.
Se richiesta fattura:
Cod. F. e P. Iva
Data
Firma

Per i residenti all'estero: pagamento anticipato (vaglia o versamento su ra. ccp)

SUPERCOMMODORE è bello, però... (ovvero suggerimenti, idee, critiche, richieste e tutto ciò che vi passa per la testa).

Nome
Cognome
ViaC.A.P.
CittàTel.

SUPERCOMMODORE

INPUT/OUTPUT

☐ COMPRO ☐ VENDO ☐ CAMBIO
☐ VIC 20 ☐ C 16 ☐ C64 ☐ PERIF. ☐ SOFTWARE

La rubrica INPUT/OUTPUT è gratuita ed
è riservata ai clienti J.Leaff. Chi desidera
praticare, vendere o cambiare hardware o
software può inviare il tagliando a J.Leaff
- Viale Realelli, 5 - 20124 MILANO

Nome
Cognome
ViaC.A.P.
CittàTel.

**Dietro un bambino
maltrattato
c'è sempre
una famiglia in crisi.**

È impossibile stabilire con esattezza quanti bambini in Italia subiscono violenze. Tanti, purtroppo. Ma una cosa è certa: la violenza che oggi un bambino subisce e impara, da grande la ritorcerà contro la società. Bisogna intervenire, e subito: agendo sulle cause. Intervendo sulla crisi familiare che è sempre alla base dei maltrattamenti a un bambino.

**"Vorrei contribuire anch'io".
Per sostenere il C.A.F. vi invio**

- | | | |
|-------------------------------------|-------------------------------------|------------------------------------|
| ☐ L. 10.000 | ☐ L. 25.000 | ☐ L. 50.000 |
| ☐ L. 100.000 | ☐ L. 500.000 | ☐ L. |

☐ Tramite assegno bancario intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano, che vi invio,
unito a questo tagliando.

☐ Tramite versamento sul C.C.P. n° 22349203 intestato C.A.F.
Via V. E. Orlando 15, 20142 Milano.

Nome e Cognome

Indirizzo

CAP.....Città.....

Provincia

Attendo: ricevuta e tesserino di "Sostenitore" del C.A.F.

**Segnaliamo
ai Servizi Sociali
del Territorio i casi
di maltrattamento.**

La prima cosa da fare: segnalare ogni caso di maltrattamento di cui siamo a conoscenza o di cui abbiamo il fondato sospetto. Telefonate ai Servizi Sociali del Territorio dal lunedì al venerdì,

dalle ore 9 alle ore 12 (troverete i numeri di telefono su Tuttocittà), oppure al Tribunale per i Minorenni (chiedendo del giudice di turno) dal lunedì al sabato dalle ore 9 alle ore 13, telefono 4988001/2/3; oppure al C.A.F., a questi numeri di telefono 02/8265051/2, che rispondono 24 ore su 24, 7 giorni su 7.

Sosteniamo il C.A.F.

Il C.A.F. è il primo Centro di Aiuto al Bambino Maltrattato e alla Famiglia in Crisi, sorto in Italia con lo scopo di intervenire nei casi di maltrattamento.

Oggi, nella sua nuova sede più grande e attrezzata, accoglie provvisoriamente i bambini maltrattati, contemporaneamente svolge un'azione terapeutica sulla famiglia per ripristinare l'equilibrio e reintegrare il bambino nel suo ambiente.

Il C.A.F. è un'istituzione privata che opera in collaborazione con i Servizi Sociali, con i supporti messi gratuitamente a disposizione dal Comune di Milano, e con i contributi finanziari volontari dei cittadini.

Contribuisci anche tu, utilizzando il tagliando. È un gesto necessario, per i bambini maltrattati, per te che vuoi fare qualcosa di concreto contro la violenza.

**Spazio gratuito
offerto da J.soft**

C.A.F.



**CENTRO DI AIUTO
AL BAMBINO MALTRATTATO
E ALLA FAMIGLIA IN CRISI**

**UN BAMBINO
MALTRATTATO OGGI
SARÀ UN UOMO
VIOLENTO DOMANI.
FACCIAMO QUALCOSA
PRIMA.**

Dalla grande edicola Jackson

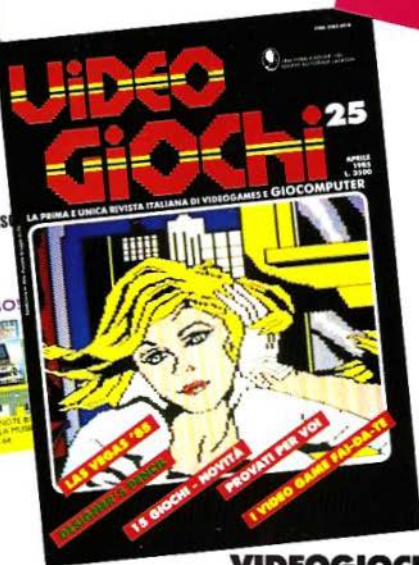
Tutto sull'hobby e home computer



STRUMENTI MUSICALI

In questo numero:

Speciale equalizzatori
Gibson Explorer
Los Angeles guitar show
Novità Francoforte '85



VIDEOGIOCHI

In questo numero:

Las Vegas '85
Designer's pencil
15 giochi-novità
I Video game fai-da-te



HOME COMPUTER

In questo numero:

Spectravideo
728 MSX
Ghostbuster
per CBM 64
Un adventure
per il QL
Match day
per Spectrum
C 16 contro VIC 20



Strumenti Musicali/Video Giochi/Home Computer
sono pubblicazioni firmate:

GRUPPO EDITORIALE JACKSON

via Rosellini, 12-20124 Milano